Master Thesis at the Computer Science Department

of the Freie Universität Berlin

Human-Centered Computing (HCC)

You shall not publish: Edit filters on English Wikipedia

Lyudmila Vaseva

Supervisor and first examiner: Prof. Dr. Claudia Müller-Birn

Second Examiner: Prof. Dr. Lutz Prechelt

Berlin, 25.07.2019

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den June 3, 2020

<Name>

Abstract

The present thesis offers an initial investigation of a previously unexplored by scientific research quality control mechanism of Wikipedia—edit filters. It is analysed how edit filters fit in the quality control system of English Wikipedia, why they were introduced, and what tasks they take over. Moreover, it is discussed why rule based systems like these seem to be still popular today, when more advanced machine learning methods are available. The findings indicate that edit filters were implemented to take care of obvious but persistent types of vandalism, disallowing these from the start so that (human) resources can be used more efficiently elsewhere (i.e. for judging less obvious cases). In addition to disallowing such vandalism, edit filters appear to be applied in ambiguous situations where an edit is disruptive but the motivation of the editor is not clear. In such cases, the filters take an "assume good faith" approach and seek via warning messages to guide the disrupting editor towards transforming their contribution to a constructive one. There are also a smaller number of filters taking care of haphazard maintenance tasks—above all tracking a certain bug or other behaviour for further investigation. Since the current work is just a first exploration into edit filters, at the end, a comprehensive list of open questions for future research is compiled.

Zusammenfassung

Die vorliegende Arbeit bietet eine erste Untersuchung eines bisher von der Wissenschaft unerforschten Qualitätskontrollmechanismus' von Wikipedia: Bearbeitungsfilter ("edit filters" auf Englisch). Es wird analysiert, wie sich Bearbeitungsfilter in das Qualitätssicherungssystem der englischsprachigen Wikipedia einfügen, warum sie eingeführt wurden und welche Aufgaben sie übernehmen. Darüberhinaus wird diskutiert, warum regelbasierte Systeme wie dieses noch heute beliebt sind, wenn fortgeschrittenere Machine Lerning Methoden verfügbar sind. Die Ergebnisse deuten darauf hin, dass Bearbeitungsfilter implementiert wurden, um sich um offensichtliche, aber hartnäckige Sorten von Vandalismus zu kümmern. Die Motivation der Wikipedia-Community war, dass wenn solcher Vandalismus von vornherein verboten wird, (Personal-)Ressourcen an anderen Stellen effizienter genutzt werden können (z.B. zur Beurteilung weniger offensichtlicher Fälle). Außerdem scheinen Bearbeitungsfilter in uneindeutigen Situationen angewendet zu werden, in denen eine Bearbeitung zwar störend ist, die Motivation der editierenden Person allerdings nicht klar als boshaft identifiziert werden kann. In solchen Fällen verinnerlichen die Filter Wikipedias "Geh von guten Absichten aus" Richtlinie und versuchen über Warnmeldungen einen konstruktiven Beitrag anzuleiten. Es gibt auch eine kleinere Anzahl von Filtern, die sich um vereinzelte Wartungsaufgaben kümmern. Hierunter fallen die Versuche, einen bestimmten Bug nachzuvollziehen oder ein anderes Verhalten zu verfolgen, um es dann weiter untersuchen zu können. Da die aktuelle Arbeit nur ein erster Einblick in Wikipedias Bearbeitungsfilter darstellt, wird am Ende eine umfassendere Liste mit offenen Fragen für die zukünftige Erforschung des Mechanismus' erarbeitet.

Contents

List of Figures

List of Tables

1. Introduction

In May 2014 the US American magasine *The New Yorker* published a story called “How a Raccoon Became an Aardvark” in its column “Annals of Technology” [Ran14]. It tells an anecdote about a student from New York who, some six years before, edited the Wikipedia article on “coati” (a member of the raccoon family native to South and Central America) to state that the coati is “also known as [...] Brazilian aardvark” [Wik08].

This simple action is a mundane example of how Wikipedia works: Anyone can edit and small contribution by small contribution the world’s largest knowledge base is created. Except, the student made the whole thing up and published on Wikipedia an inside joke he had with his brother on their holiday trip to Brazil. Unsourced pieces of information are not supposed to survive long on Wikipedia and he thought that the edit would be swiftly deleted. Fast-forward to 2014, not only had this part of the “coati” entry not changed, but it cited a 2010 article by the newspaper the *Telegraph* as evidence [Wik14]. In the meantime, apparently several newspapers, a YouTube video, and a book published by the University of Chicago [Hen13] claimed that the coati was known as Brazilian aardvark. It proved not trivial to erase the snippet from Wikipedia since there were all these other sources affirming the statement. By then, it was not exactly false either: The coati *was* known as “Brazilian aardvark”, at least on the Internet.

Now, despite accounts that Wikipedia seems to be similarly accurate and more complete than the online version of encyclopedia Britannica [CDFN12], [Gil05] stories like the one above are precisely why it is still maintained that information on Wikipedia cannot be trusted, or used as a serious bibliographic reference.

The Wikipedian community is well-aware of their project’s poor reliability reputation and has a long standing history of quality control processes. Not only hoaxes, but profanities, malicious vandals, and spammers have been there since the very beginning and their numbers have increased with the rise of the project to prominence. At the latest, with the exponential surge in the numbers of users and edits around 2006, the community began realising that they needed a more automated means for quality control. The same year, the first anti-vandal bots were implemented, followed by semi-automated tools facilitating revision verification such as Twinkle [Wik19bd] (in 2007) and Huggle [Wik19ak] (in the beginnings of 2008). In 2009, yet another mechanism dedicated to quality control was introduced. Its core developer, Andrew Garrett, known on Wikipedia as User:Werdna [Wik19bf], has called it “abuse filter”, and according to EN Wikipedia’s community newspaper, The Signpost, its purpose was to “allow [...] all edits to be checked against automatic filters and heuristics, which can be set up to look for patterns of vandalism including page move vandalism and juvenile-type vandalism, as well as common newbie mistakes" [Sig09]. This mechanism is the focus of the current project.

1.1. Subject and Context

The present thesis can be embedded in the context of (algorithmic) quality-control on Wikipedia and in the more general research area of algorithmic governance [DHN+17], [MBDH13]. There is a whole ecosystem of actors struggling to maintain the anyone-can-edit encyclopedia as accurate and free of malicious content as possible. The focus of this work are edit filters, the mechanism initially introduced by User:Werdna under the name of "abuse filters", previously unexplored by the scientific community. The goal of this project is to better understand the role of edit filters in the vandal fighting network of humans, bots, semi-automated tools, and the Wikipedian machine learning framework ORES. After all, edit filters were introduced to Wikipedia at a time when the majority of the aforementioned mechanisms already existed and were involved in quality control [1].

1.2. Contributions

The aim of this work is to find out why edit filters were introduced on Wikipedia and what role they assume in Wikipedia's quality control ecosystem since there is a gap in the academic research on the topic. Further, it is analysed what tasks are taken over by filters and—as far as practicable—tracked how these tasks have evolved over time (are there changes in type, numbers, etc.?). Moreover, it is discussed why a classic rule based system such as the filters is still operational today when more sophisticated machine learning (ML) approaches exist. Since this is just an initial discovery of the features, tasks and repercussions of edit filters, a framework for future research is also offered.

To this end, a three path approach is pursued. Firstly, the academic contributions on Wikipedia's quality control mechanisms are reviewed in order to gather a better understanding of the different quality control mechanisms, their tasks, and the challenges they face. Then, the documentation of the MediaWiki AbuseFilter extension is studied, together with the guidelines for its use, various noticeboards, and discussion archives prior to its introduction in an attempt to understand how and why filters were introduced and how they function. Thirdly, I look into the filters implemented on English Wikipedia [2]

[1] Edit filters were introduced in 2009. The page of the semi-automated tool Twinkle [Wik19bd] was created in January 2007, the one of the tool Huggle [Wik19ak]—in the beginning of 2008. Bots have been around longer, but first records of vandal fighting bots come from 2006.

[2] Throughout the work, the abbreviated form "EN Wikipedia" is used to denote the English language version of Wikipedia.

themselves, as well as their log data in order to determine what they actually do.

First results show that edit filters were implemented to take care of obvious but persistent types of vandalism such as mass moves of pages to nonsense URLs. The community was willing to disallow this kind of edits from the very start, reasoning that the efforts spent to clean up such cases can be used more efficiently elsewhere (i.e. for judging whether less obvious cases were malicious or not). In addition to disallowing such vandalism, edit filters appear to be applied in ambiguous situations where an edit in itself is disruptive but the motivation of the editor is not clear. For example, deleting the entire content of a page could be malignant, but it could also be the result of a new editor not familiar with proper procedures for deleting or moving pages. In such cases, the filters take an "assume good faith" approach and seek via warning messages to guide the disrupting editor towards transforming their contribution into a constructive one: In the page blanking example, a warning contains links to the documentation for redirects and the Articles for Deletion process [Wik19j], and advises the editor to revert the page to the last uncompromised version in case it has been vandalised, and to use the sandbox for test edits. There are also a smaller number of filters which take care of various maintenance tasks—above all tracking a certain bug or other behaviour for further investigation. Since the current work is just a first exploration into edit filters, at the end, a comprehensive list of open questions for future research is compiled.

1.3. Structure

This thesis is organised in the following manner: Chapter 2 situates the topic in the academic discourse by examining the role of different quality control mechanisms on Wikipedia hitherto studied by the scientific community. In chapter 3, I present the methodological frameworks on which this research is based. Next, the edit filter mechanism in general is described: How and why it was conceived, how it works and how it is embedded in Wikipedia's quality control ecosystem (chapter 4). A detailed analysis of the current state of all implemented edit filters on English Wikipedia is presented in chapter 5. Chapter 6 discusses the findings and limitations of the present work, as well as directions for future investigations. Finally, the research is wrapped up in chapter 7.

2. Quality-Control Mechanisms on Wikipedia

The present chapter studies the scientific literature on Wikipedia's quality control mechanisms in order to better understand the role of edit filters in this ecosystem.

Before 2009, academic studies on Wikipedia tended to ignore algorithmic agents altogether. The number of their contributions to the encyclopedia was found to be low and therefore their impact was considered insignificant [KCP+07]. This has gradually changed since around 2009 when the first papers specifically dedicated to bots (and later semi-automated tools such as Huggle and Twinkle) were published. In 2010, Geiger and Ribes insistently highlighted that the scientific community could no longer neglect these mechanisms as unimportant or noise in the data [GR10].

For one, the mechanisms' relative usage has continued to increase since they were first introduced [Gei09]. What is more, Geiger and Ribes argue, the algorithmic quality control mechanisms change the system not only in a matter of scale (using bots/tools is faster, hence more reverts are possible) but in a matter of substance: the very way everything interacts with each other is transformed [GR10]. On the grounds of quality control specifically, the introduction of algorithmic mechanisms was fairly revolutionary: They enabled efficient patrolling of articles by users with little to no knowledge about the particular topic. Thanks to Wikipedia's idiosyncratic software architecture, this is possible even in the most "manual" quality control work (i.e. using watchlists to patrol articles): Representing information changes via diffs allows editors to quickly spot content that deviates from its immediate context [GR10].

Others were worried it was getting increasingly untransparent how the encyclopedia functions and not only "[k]eeping traces obscure help[ed] the powerful to remain in power" [FG12], but entry barriers for new users were gradually set higher [HGMR13]: They had to learn to interact with a myriad of technical tools, learn wiki syntax, but also navigate their ground in a complex system with a decentralised socio-technical mode of governance [Gei17]. Ford and Geiger even cite a case in which an editor was not sure whether a person deleted their articles or a bot [FG12].

Quality control mechanisms on Wikipedia can be categorised into following three groups according to their level of automation: fully automated, semi-automated, and manual. Fully automated tools include bots, the edit filters, which are the focus of the present thesis, and other MediaWiki's features such as the mechanism for page protection [Med19e] (which allows restricting editing of a particular page to certain usergroups), and title [Med19g] and spam [Med19f] blacklists which operate on regular expression basis to disallow specific titles or publication of some links perceived as spam. There is also the automatically functioning Wikipedian machine learning framework ORES [ORE19] which computes quality scores per article or revision. ORES has a somewhat different status compared to the other technologies listed here: It is a meta tool whose scores can be employed by other mechanisms. Semi-automated tools still need some sort of a user interaction/confirmation in order to operate. In this category fall the tools Huggle [Wik19ak], Twinkle [Wik19bd] and STiki [Wik19bb]. There are also some semi-automated bots (although most prominent anti-vandalism bots discussed here are fully automated). Manual quality control work is done by human editors without the help of any particular software program.

The following sections discuss what the scientific community already knows about the different mechanisms in order to be able to situate edit filters in Wikipedia's quality control ecosystem.

2.1. Automated

2.1.1. Bots

According to the literature, bots constitute the first "line of defence" against malicious edits [GH13]. They are also undoubtedly the quality control mechanism studied most in-depth by the scientific community.

Geiger and Ribes [GR10] define bots as "fully-automated software agents that perform algorithmically-defined tasks involved with editing, maintenance, and administration in Wikipedia" [1].

Different aspects of bots and their involvement in quality control have been investigated: In the paper referenced above, the researchers employ their method of trace ethnography (more on it in chapter 3) to follow a disrupting editor around Wikipedia and comprehend the measures taken in collaboration by bots (ClueBot [Wik19m] and HBC AIV helperbot7 [Wik19aj]) as well as humans using semi-automated tools (Huggle [Wik19ak] and Twinkle [Wik19bd]) up until they achieved that the malicious editor in question was banned [GR10]. Halfaker and Riedl offer a historical review of bots and semi-automated tools and their involvement in vandal fighting [HR12], assembling a comprehensive list of tools and touching on their working principle (rule vs. machine learning based). They also develop a bot taxonomy classifying bots in one of the following three groups according to their task area: content injection, monitoring or curating; augmenting MediaWiki functionality; and protection from malicious activity. In [GH13], Geiger and Halfaker conduct an in-depth analysis of ClueBot NG, ClueBot's machine learning based successor, and its place within Wikipedia's vandal fighting infrastructure concluding that quality

[1] Not all bots are completely automated: There are batch scripts started manually and there are also bots that still need a final click by a human. However, the ones the present work focuses on—the rapid response anti-vandalism agents such as ClueBot NG [Wik19n] and XLinkBot [Wik19bj]—work in a fully automated fashion.

control on Wikipedia is a robust process and most malicious edits eventually get reverted even with some of the actors (temporaly) inactive, although at a different speed. They discuss the mean times to revert of different mechanisms, their observations coinciding with figure 2.1, and also comment on the (un)realiability of external infrastructure bots rely upon (run on private computers, which causes downtimes).

Further bots involved in vandal fighting (besides ClueBot [GR10] and ClueBot NG [GH13], [HR12]) discussed by the literature include: XLinkBot (which reverts edits containing links to domains blacklisted as spam) [HR12], HBC AIV Helperbots (responsible for various maintenance tasks which help to keep entries on the Administrator intervention against vandalism (AIV) dashboard up-to-date) [HR12], [GR10], MartinBot [Wik19ap] and AntiVandalBot [Wik19h] (one of the first rule-based bots which detected obvious cases of vandalism) [HR12], DumbBOT [Wik19q] and EmausBot [Wik19ag] (which do batch cleanup tasks) [GH13].

Very crucial for the current analysis will also be Livingstone's observation in the preamble to his interview with the first large scale bot operator Ram-man that "[i]n the Wikimedia software, there are tasks that do all sorts of things [...]. If these things are not in the software, an external bot could do them. [...] The main difference is where it runs and who runs it" [Liv16]. This thought is also scrutinised by Geiger [Gei14] who examines in detail what the difference and repercussions are of code that is part of the core software and code that runs alongside it (such as bots) which he calls "bespoke code". Geiger pictures Wikipedia as a big socio-technical assemblage of software pieces and social processes, often completely untransparent for an outside observer who is not able to identify the single components of this system and how they interact with one another to provide the end result to the public. He underlines that components which are not strictly part of the server-side codebase but run by various volunteers (which is well true for the most parts of Wikipedia, it is a community project) on their private infrastructure constitute the major part of Wikipedia and also that they can experience a downtime at any moment. The vital tasks they perform, such as vandalism fighting, are often taken for granted, much to their developers' aggravation.

A final aspect in the bot discussion relevant here are the concerns of the community. People have been long sceptical (and some still are) about the employment of fully automated agents such as bots within Wikipedia (some has called this fear "botophobia" [Gei11]). Above all, there is a fear of bots (especially such with admin permissions) running rampant and their operators not reacting fast enough to prevent the damage. This led to the social understanding that "bots ought to be better behaved than people" [Gei11] which still plays a crucial role in bot development today.

2.1.2. ORES

ORES [ORE19] is an API based free libre and open source (FLOSS) machine learning service “designed to improve the way editors maintain the quality of Wikipedia” [HT15] and increase the transparency of the quality control process. It uses learning models to predict a quality score for each article and edit based on edit/article quality assessments manually assigned by Wikipedians. Potentially damaging edits are highlighted, which allows editors who engage in vandal fighting to examine them in greater detail. The service was officially introduced in November 2015 by Aaron Halfaker[2] (principal research scientist at the Wikimedia Foundation [3]) and Dario Taraborelli[4] (Head of Research at Wikimedia Foundation at the time) [HT15]. Its development is ongoing, coordinated and advanced by Wikimedia’s Scoring Platform team. Since ORES is API based, in theory a myriad of services can be developed that use the predicted scores, or new models can be trained and made available for everyone to use. As already mentioned, the tool has a meta status, since it does not fight vandals on its own, but rather it can be employed by other mechanisms for determining the probability that a particular edit is disruptive. The Scoring Platform team reports that popular quality control tools such as Huggle (see next section) have already adopted ORES scores for the compilation of their queues [HT15]. What is unique about ORES is that all the algorithms, models, training data, and code are public, so everyone (with sufficient knowledge of the matter) can scrutinise them and reconstruct what is going on. Halfaker and Taraborelli express the hope that ORES would help hone quality control mechanisms on Wikipedia, and by decoupling the damage prediction from the actual decision how to deal with an edit make the encyclopedia more welcoming towards newcomers. This last aim is crucial, since there is a body of research demonstrating how reverts in general [HKR11] and reverts by (semi-)automated quality control mechanisms in particular drive new editors away [HGMR13]. Present authors also signal that these tools still tend to reject the majority of newcomers’ edits as made in bad faith. The researchers also warn that wording is tremendously important for the perception of edits and people who authored them: labels such as “good” or “bad” are not helpful.

2.1.3. Page Protection, TitleBlacklist, SpamBlacklist

Page protection is a MediaWiki template-based functionality which allows administrators to restrict edit access to a particular page temporarily (the most

[2] https://wikimediafoundation.org/role/staff-contractors/

[3] The Wikimedia Foundation is a non-profit organisation dedicated to collecting and disseminating free knowledge [Wik19a]. Beside Wikipedia, it provides and maintains the infrastructure for a family of projects such as Wikimedia Commons (a collection of freely usable media), Wiktionary (a free dictionary), or Wikidata (a free structured knowledge base) [Wik19b].

[4] http://nitens.org/taraborelli/cv

common periods are 7 and 30 days) or permanently [Wik19aw], [GH17]. The mechanism is suitable for handling a higher number of incidents concerning single pages [Wik19r]. Only one study dedicated specifically to page protection on Wikipedia was found—[HS15]. In this paper, Hill and Shaw maintain that the mechanism is highly configurable: available in more than ten varieties including the most popular "full protection" (only administrators can edit) and "semi-protection" (only registered, autoconfirmed users can edit). Moreover, it is found that pages are protected for various reasons, e.g. to prevent edit warring or vandalism; to enforce a policy or the law; it is an established process to protect articles on the front page. The researchers also look into the historical development of protected pages on Wikipedia and discuss the repercussions of the mechanism for affected users [HS15]. If a user doesn't have the permissions needed to edit protected page, the "edit" link is simply not displayed at all.

The rule-based MediaWiki extensions TitleBlacklist [Med19g] and SpamBlacklist [Med19f] are employed for disallowing disruptive page titles or link spam. The only more extensive account found on these mechanisms discusses link spam on Wikipedia and has identified the SpamBlacklist as the first mechanism to get activated in the spam removal pipeline [WCV+11].

2.2. Semi-Automated

Semi-automated quality control tools are similar to bots in the sense that they provide automated detection of potential low-quality edits. The difference however is that with semi-automated tools humans do the final assessment and decide what happens with the edits in question.

There is a scientific discussion of several tools: Huggle [Wik19ak], which is probably the most popular and widely used one, is studied in [GH13], [HR12], and [GR10]. Another very popular tool, Twinkle [Wik19bd], is commented on by [GH13], [GR10], and [HGMR13]. STiki [Wik19bb] is presented by its authors in [WKL10] and also discussed by [GH13]. Various older (and partially inactive) applications are mentioned by the literature as well: Geiger and Ribes [GR10] touch on Lupin's Anti-vandal tool [Wik19ao], Halfaker and Riedl talk about VandalProof [HR12].

Some of these tools are more automated than others: Huggle and STiki for instance are able to revert an edit, issue a warning to the offending editor, and post a report on the AIV dashboard (if the user has already exhausted the warning limit) upon a single click. The javascript based browser extension Twinkle on the other hand adds contextual links to other parts of Wikipedia which facilitates fulfillment of particular tasks such as rollback multiple edits, report problematic users to AIV, or nominate an article for deletion [GR10]. The main feature of Huggle and STiki is that they both compile a central queue of potentially harmful edits for all their users to check. The difference between both programs are the heuristics they use for their queues: By default, Huggle sends edits by users with warnings on their user talk page to the top of the queue, places edits by IP editors higher and ignores edits made by bots and other Huggle users altogether[GR10]. In contrast, STiki relies on the "spatio-temporal properties of revision metadata" [WKL10] for deciding the likelihood of an edit to be vandalism. Huggle's queue can be reconfigured, however, some technical savvy and motivation is needed for this and thus, as [GR10] warn, it makes certain paths of action easier to take than others. Another common trait of both programs is that as a standard, editors need the "rollback" permission in order to be able to use them [HR12].

Some critique that has been voiced regarding semi-automated anti-vandalism tools compares these to massively multiplayer online role-playing games (also known as MMORPGs) [HR12]. The concern is that some of the users of said tools see themselves as vandal fighters on a mission to slay the greatest number of monsters (vandals) possible and by doing so to excell in the ranks [5]. This is for one a harmful way to view the project, neglecting the "assume good faith" guideline [Wik19k] and also leads to such users seeking out easy to judge instances from the queues in order to move onto the next entry more swiftly and gather more points leaving more subtle cases which really require human judgement to others.

Transparency wise, one can criticise that the heuristics they use to compile the queues of potential malicious edits in need of attention are oftentimes obfuscated by the user interface and so the editors using them are not aware why exactly these and not other edits are displayed to them. The heuristics to use are configurable to an extent, however, one needs to be aware of this option [GR10].

2.3. Manual

For completion, it should be noted at this point that despite the steady increase of the proportion of fully and semi-automated tools usage for fighting vandalism [Gei09], some of the quality control work is still done "manually" by human editors. These are, on one hand, editors who use the "undo" functionality from within the page's revision history. On the other hand, there are also editors who engage with the classic encyclopedia editing mechanism (click the "edit" button on an article, enter changes in the dialog which opens, write an edit summary for the edit, click "save") rather than using further automated tools to aid them. When Wikipedians use these mechanisms for vandalism fighting, oftentimes they haven't noticed the vandalising edits by chance but rather have been actively watching the pages in question via the so-called watchlists [AH18]. This also gives us a hint as to what type of quality control work humans take over: less obvious and less rapid, requiring more complex judgement [AH18]. Editors who patrol pages via watchlists often have some relationship to/deeper expertise on the topic.

[5] STiki actually has a leader board: https://en.wikipedia.org/w/index.php?title=Wikipedia:STiki/leaderboard&oldid=905145147

2.4. Conclusion

For clarity, the various aspects of algorithmic quality control mechanisms learnt by studying related works are summarised in table 2.1. Their work can be fittingly illustrated by figure 2.1, proposed in a similar fashion also by [AH18]. What strikes about this diagram is that it foregrounds the temporal dimension of quality control work done on Wikipedia demonstrating that as a general rule bots are the first mechanisms to intercept a potentially harmful edit, less obviously disruptive edits are often caught by semi-automated quality control tools and really subtle cases are uncovered by manually reviewing humans or sometimes not at all.

One thing is certain: So far, on grounds of literature review alone, it remains unclear what the role of edit filters is. The mechanism is ostentatiously missing from the studied accounts. In the remainder of the current thesis, I try to remedy this gap in research by exploring following questions:
Q1: What is the role of edit filters among existing algorithmic quality-control mechanisms on Wikipedia (bots, semi-automated tools, ORES, humans)?
Q2: Edit filters are a classical rule-based system. Why are they still active today when more sophisticated ML approaches exist?
Q3: Which type of tasks do filters take over?
Q4: How have these tasks evolved over time (are there changes in the type, number, etc.)?

In order to be able to answer them, various Wikipedia's pages, among other things policies, guidelines, documentation and discussions, are studied in chapter 4 and filter data from the English Wikipedia is analysed in chapter 5. But first, chapter 3 introduces the applied methodology.

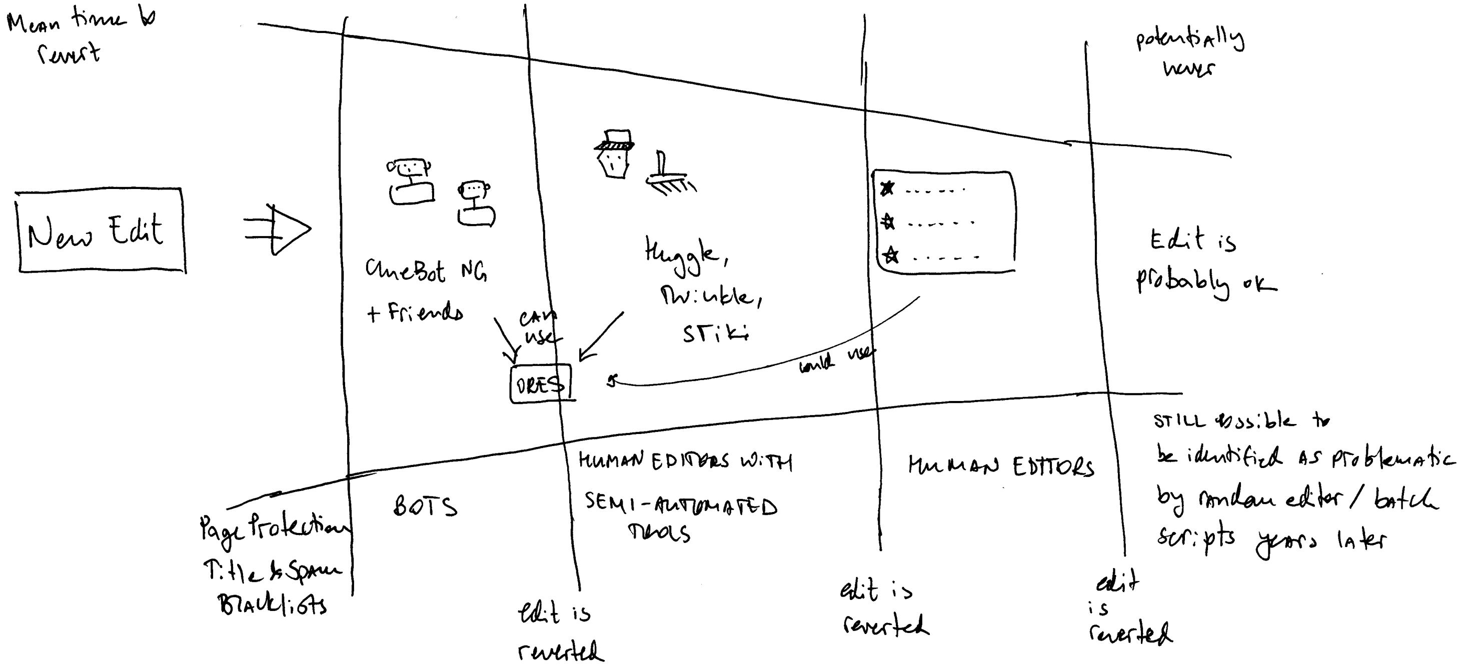


Figure 2.1.: State of the scientific literature: edit filters are missing from the quality control ecosystem

	Page protection	Blacklists	Bots	Semi-Automated tools	ORES
Properties	per page	rule-based	rule/ML based	rule/ML based	ML framework
	part of MediaWiki	part of MediaWiki	run on user's infrastructure ("bespoke code")	extra infrastructure	not used directly, can be incorporated in other tools
	MediaWiki is open source		no requirement for code to be public	most popular are open source (but not a hard requirement)	open source
				heuristics obfuscated by the interface	
	one cannot edit at all	trigger before an edit is published	latency varies (ClueBot NG needs only a couple of seconds)	generally higher latency than bots	
			mostly single dev/operator (recently: bot frameworks)	few devs	few devs

Necessary permissons	admin access	admin access	no special rights needed (except for admin bots) bot gets a “bot flag”	*rollback* permission needed	
People involved	admins	admins (anyone can report problems)	bot operators	editors with *rollback* permission	mostly Scoring Platform team
Hurdles to participate		understand regexes	get approval from the BAG programming knowledge, understand APIs, ...	get a *rollback* permission get familiar with the tool	understand ML
Concerns			“botophobia”	gamification	general ML concerns: hard to understand
Areas of application	problems with single page	abusive titles/ link spam	mostly obvious vandalism	less obvious cases that require human judgement	

Table 2.1.: Wikipedia’s algorithmic quality control mechanisms in comparison

3. Methods

This chapter provides the theoretical background for the study of edit filters. I make use of trace ethnography, described in the following section, in the study of documentation and discussion archives conducted in chapter 4 in order to understand the role of edit filters in the quality control ecosystem of English Wikipedia. The emergent coding introduced in section 3.2, combined with trace ethnography, is employed in chapter 5 for determining what tasks edit filters take care of.

The whole work tries to adhere to the principles of open science and reproducible research. According to the definition of Bartling and Friesike provided in their book *Opening Science* [BF14], open science is primarily characterised, unsurprisingly, by its openness. There is an open communication of the methods and results in every stage of the research project, allowing, importantly, for an easier disclosure of negative results. The code for all data processing and computational analyses I have done, as well as other artefacts I have used or compiled have been openly accessible in the project's repository since the beginning of the present research [Git19] and can be re-used under a free license. Anyone interested can follow the process and/or use the data or scripts to verify my computations or run their own and thus continue this research along one of the directions suggested in section 6.6 or in a completely new one.

3.1. Trace Ethnography

Trace ethnography constitutes the main theoretical framework for the analysis presented in chapters 4 and 5. The concept was coined by Geiger and Ribes in their 2010 paper "The work of sustaining order in Wikipedia: the banning of a vandal" [GR10] and introduced in detail in a 2011 article [GR11] by the same authors. They define trace ethnography as a methodology which "combines the richness of participant-observation with the wealth of data in logs so as to reconstruct patterns and practices of users in distributed sociotechnical systems". It extends classic documentary ethnography which can rely on any written artefact such as archive records, diaries, manuals and handbooks, correspondence, standards, protocols, or trading records, by the extensive use of logs and other records generated by digital environments in an automated manner. The method is supposedly especially useful for research in the cited distributed sociotechnical systems (such as Wikipedia), since there direct participants observation is often impractical, costly and tend to miss phenomena which manifest themselves in the communication between spatially separated sites rather than in the single location.

In [GR10] the scholars use documents and document traces: MediaWiki revision data, more specifically—edit summary fields of the single revisions and markers left programatically within the edit summaries; documentation of semi-automated software tools; and even use the tools (Huggle and Twinkle) themselves to observe what traces these leave; in order to reconstruct quite exactly individual strands of actions and comprehend how different agents on Wikipedia work together towards the blocking of a single malicious user. They refer to "turn[ing] thin documentary traces into "thick descriptions" of actors and events" and "inverting traces" in order to reconstruct sequences of events and actions. What is more, these traces are used by Wikipedians themselves in order to do their work efficiently: For example, after seeing an entry on the Administrator Intervention against Vandalism noticeboard [Wik19f], an admin would probably look up the latest actions of a user, as well as check their user talk page for recent warnings, before deciding to block the user in question. Geiger and Ribes underline the importance of insider knowledge when reconstructing actions and processes based on the traces, the need for "an ethnographic understanding of the activities, people, systems, and technologies which contribute to their production".

They alert that via trace ethnography only that can be observed which is recorded by the system and that records are always incomplete. This consideration is elaborated on in more detail in [GH17], where Geiger and Halfaker make the point that "found data" generated by a system for a particular purpose (e.g. revision history whose purpose is to keep a track of who edited what when and possibly revert (to) a particular revision) is rarely ideally fitting as a dataset to answer the particular research question of a scientist. The importance of interpreting data in their corresponding context and the pitfalls of tearing analysis out of context are also underlined by Charmaz in [Cha06]. She cites intersecting data from multiple sources/of different types as a possible remedy for this problem.

Last but not least, Geiger and Ribes [GR11] also warn of possible privacy breaching through thickening traces: Although the records they use to reconstruct paths of action are all open, the thick descriptions compiled can suddenly expose a lot of information about single users which never existed in this form before and the individuals concerned never gave their informed consent for their data being used this way.

3.2. Emergent Coding

In order to gain a detailed understanding of what edit filters are used for on English Wikipedia, in chapter 5 all filters are scrutinised and labeled via a technique called emergent coding. Coding is the process of labeling data in a systematic fashion in an attempt to comprehend it. It is about seeking patterns in data and later—trying to understand these patterns and the relationships between them. Emergent coding is one possible approach for making sense of data in content analysis [Ste01]. Its key characteristic is letting the codes emerge during the process contrasted to starting with a set of preconceived codes (also known as “a priori codes”). Scholars regard this as useful because that way the danger of trying to press data into predefined categories while potentially overlooking other, better fitting codes is reduced [Cha06, p.17]. Instead, the codes stem directly from observations of the data.

Traditionally in content analysis, there are at least two researchers involved in an emergent coding process. During an initial examination of the data, they independently come up with preliminary codes [1] which are then compared and discussed until a consolidated code book is developed. Then, all researchers involved use this code book to—again independently—label the data. At the end, their labelings are compared and the reliability of the coding is verified. If the results don’t reach a pre-defined agreement level, differences are discussed and previous steps are repeated.

It has to be mentioned here that for the present project only one coder was available—me. Therefore, unfortunately, the prescribed validation steps could not be realised. The exact form of the applied coding process and the resulting limititations are described in sections 5.2 and 6.5 respectively.

[1] I use the words “codes”, “labels”, “tag”, and “categories” interchangeably.

4. Edit Filters As Part of Wikipedia's Socio-Technical Infrastructure

"Abuse Filter is enabled" reads the title of one of the eight stories of the 23 March 2009 issue of English Wikipedia's community newspaper, The Signpost [Sig09]. "The extension allows all edits to be checked against automatic filters and heuristics, which can be set up to look for patterns of vandalism including page move vandalism and juvenile-type vandalism, as well as common newbie mistakes," the article proclaims.

The extension, or at least its end user facing parts, was later renamed to "edit filter" in order to not characterise false positives as "abuse" and thus alienate good faith editors striving to improve the encyclopedia [Wik19r], [Wik19af].

The aim of this chapter is to understand how edit filters work, who implements and runs them and above all, how and why they were introduced in the first place and what the qualitative difference is between them and other algorithmic quality control mechanisms. The analysed data is presented in the following section. Section 4.2 defines what an edit filter is. The AbuseFilter MediaWiki extension is introduced in section 4.3. After this common understanding of the state of the art of edit filters has been established, section 4.4 looks back and traces the historical debate that led to the filters' introduction. Sections 4.5 and 4.6 take respectively an internal (edit filter managers') and an external (all other users') perspective towards the edit filters. The findings are then compared with the results from chapter 2 and discussed in section 4.7.

4.1. Data

The foundations for the present chapter lie in EN Wikipedia's policies and guidelines. Following pages were analysed in depth:
https://en.wikipedia.org/wiki/Wikipedia:Edit_filter
https://en.wikipedia.org/wiki/Wikipedia:Edit_filter/Documentation
https://en.wikipedia.org/wiki/Wikipedia:Edit_filter/Instructions
https://en.wikipedia.org/wiki/Wikipedia:Edit_filter_noticeboard
https://en.wikipedia.org/wiki/Wikipedia_talk:Edit_filter/Archive_1

4.2. Edit Filter Definition

Every edit filter defines a pattern [1] against which every edit made to Wikipedia is checked. If there is a match, the edit in question is logged and potentially, additional actions such as tagging the edit summary, issuing a warning or disallowing the edit are invoked. Both the patterns and the possible edit filter actions are investigated in greater detail in the following sections.

According to EN Wikipedia's own definition, an edit filter is "a tool that allows editors in the edit filter manager group to set controls mainly to address common patterns of harmful editing" [Wik19r].

A couple of keywords arouse interest here: Who is in the edit filter manager group and how did they become part of it? What controls exactly can be set? What does "mainly" mean, are there other patterns addressed? And what are the patterns of harmful editing addressed by the filters?

At least the "mainly" question is swiftly answered by the paragraph itself, since there is a footnote stating that "[e]dit filters can and have been used to track or tag certain non-harmful edits, for example addition of WikiLove" [Wik19r]. The controls that can be set are looked into in the sections that follow. The edit filter manager group and its members are discussed in section 4.5.2 and the patterns of harmful editing (as well as some further non-harmful edit patterns) are inspected in detail in the next chapter.

4.2.1. Example of a Filter

For illustration purposes, let us have a closer look at what a single edit filter looks like. Edit filter with ID 365 is public [2] and currently enabled (as of 30 June 2019). This means the filter is working and everyone interested can view the filter's details. Its description reads "Unusual changes to featured or good content". The filter pattern is:

```
"page_namespace == 0 &
!(""confirmed"" in user_groups) &
old_size > 20000 & (
    ""#redirect"" in lcase(added_lines) |
    edit_delta < -15000 |
    edit_delta > 15000
) &
old_wikitext rlike
""\{\{([Ff]eatured|[Gg]ood)\s?article\}\}"""
```

And the currently configured filter actions are: "disallow".

[1] These patterns consist of one or more conditions, e.g. matching the edit's content against a regular expression or checking the usergroups of the contributing editor.

[2] There are also private (hidden) filters. The distinction is discussed in more detail in sections 4.4 and 5.3.2.

So, if a user whose status is not confirmed [3] yet tries to edit a page in the article namespace which contains “Featured” or “Good article” and they either insert a redirect, delete 3/4 of the content or add 3/4 on top, the edit is automatically disallowed.

Note that an edit filter editor can easily change the action of the filter. (Or the pattern, as a matter of fact.) The filter was last modified on 23 October 2018. All these details can be viewed on the filter’s detailed page [Wik19s] or on the screenshot thereof (figure 4.1) that I created for convenience.

Further information the filter detailed page displays is: number of filter hits; some statistics (the average time the filter takes to check an edit, percentage of hits and how many conditions from the condition limit it consumes [4]); comments (left by edit filter managers, generally to log and explain changes); flags (“Hide details of this filter from public view”, “Enable this filter”, “Mark as deleted”); links to last modified (with diff and user who modified it), the edit filter’s history and a tool for exporting the filter to another wiki; and actions to take when the filter’s pattern matches.

4.3. The AbuseFilter[5] Mediawiki Extension

At the end, from a technical perspective, Wikipedia’s edit filters are a MediaWiki plugin that allows every edit (and some other editor’s actions) to be checked against a specified pattern before it is published.

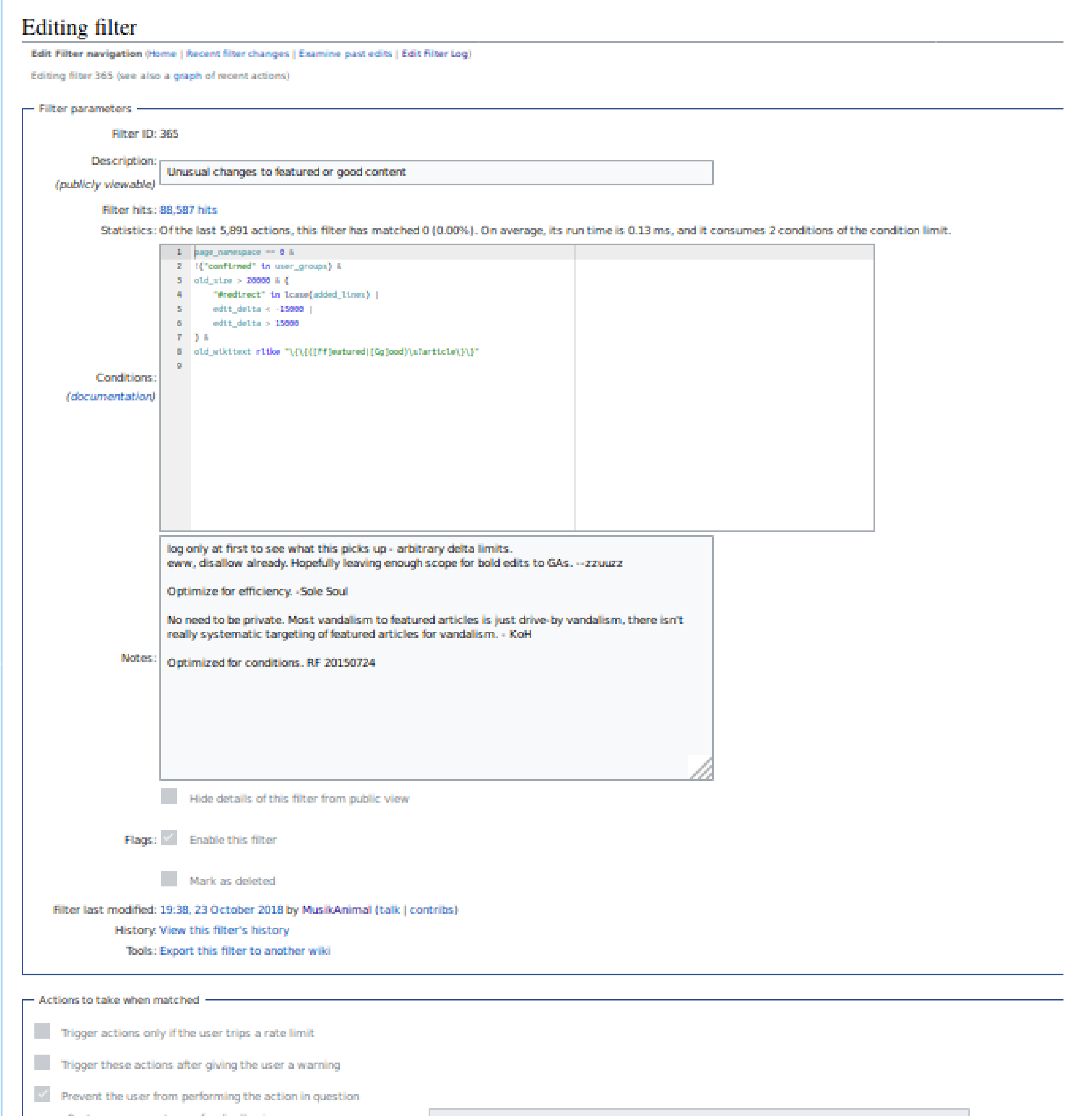
The extension introduces following database tables where all data generated by it is stored: *abuse_filter*, *abuse_filter_log*, *abuse_filter_action*, and *abuse_filter_history* [Gar19b]. *abuse_filter* contains detailed information about every filter. *abuse_filter_action* stores the currently configured actions for each filter and their corresponding parameters. Every update of a filter action, pattern, comments or other flags (whether the filter is enabled, hidden, deleted), etc. is recorded in *abuse_filter_history*. And every time a filter matches, the editor’s action that triggered it as well as further data such as the user who triggered the filter, their IP address, a diff of the edit (if it was an edit), a timestamp, the title of the page the user was looking at, etc. are logged in *abuse_filter_log*.

Most frequently, edit filters are triggered upon new edits, there are however

[3] A confirmed user can do following five things a non-confirmed user cannot: create pages; move pages; edit semi-protected pages; upload files; vote in certain elections (a different minimum edit count can be required for certain elections). An account can be explicitly confirmed, most accounts are autoconfirmed though. Generally, accounts are autoconfirmed when they have been registered for at least four days and have made a minimum of ten edits [Wik19d]. The requirements are adjustable.

[4] According to various community comments, both of these numbers are not particularly reliable and should be treated with caution [Med19b].

[5] Note that the user facing elements of this extension were renamed to “edit filter”, however the extension itself, as well as its corresponding permissions, database tables etc. still reflect the original name.



Editing filter

Edit Filter navigation (Home | Recent filter changes | Examine past edits | Edit Filter Log)

Editing filter 365 (see also a graph of recent actions)

Filter parameters

Filter ID: 365

Description: (publicly viewable) Unusual changes to featured or good content

Filter hits: 88,587 hits

Statistics: Of the last 5,891 actions, this filter has matched 0 (0.00%). On average, its run time is 0.13 ms, and it consumes 2 conditions of the condition limit.

Conditions: (documentation)

```
page_namespace == 0 &
!("confirmed" in user_groups) &
old_size > 20000 & (
    "#redirect" in lcase(added_lines) |
    edit_delta < -15000 |
    edit_delta > 15000
) &
old_wikitext rlike "\{\{([Ff]eatured|[Gg]ood)\s?article\}\}"
```

Notes:

log only at first to see what this picks up - arbitrary delta limits.
eww, disallow already. Hopefully leaving enough scope for bold edits to GAs. --zzuuzz

Optimize for efficiency. -Sole Soul

No need to be private. Most vandalism to featured articles is just drive-by vandalism, there isn't really systematic targeting of featured articles for vandalism. - KoH

Optimized for conditions. RF 20150724

Flags:
- □ Hide details of this filter from public view
- ☑ Enable this filter
- □ Mark as deleted

Filter last modified: 19:38, 23 October 2018 by MusikAnimal (talk | contribs)

History: View this filter's history

Tools: Export this filter to another wiki

Actions to take when matched

- □ Trigger actions only if the user trips a rate limit
- □ Trigger these actions after giving the user a warning
- ☑ Prevent the user from performing the action in question

Figure 4.1.: Detailed page of edit filter #365

further editor's actions that can trip an edit filter. As of 30 June 2019, these include: *edit*, *move*, *delete*, *createaccount*, *autocreateaccount*, *upload*, *stashupload*[6]. Historically, further editor's actions such as *feedback*, *gatheredit* and *moodbar* could trigger an edit filter. These are in the meantime deprecated.

When a filter's pattern is matched, beside logging the event in the *abuse_filter_log* table (the only filter action which cannot be switched off), a further filter action may be invoked as well. The plugin defines following possible filter actions: *tag*, *throttle*, *warn*, *blockautopromote*, *block*, *degroup*, *rangeblock*, *disallow*[7]. The documentation of the AbuseFilter extension provides us comprehensive definitions for these [Med19a]:

- *tag*: The contribution is tagged with a specific tag (which can be defined and styled by the edit filter manager) which then appears on Recent Changes, contributions, logs, history pages, etc. and allows aggregations of lists for dashboards and similar.

- *throttle*: The filter is activated upon the tripping of a rate limit. Configurable parameters are the allowed number of actions, the period of time in which these actions must occur, and how those actions are grouped. Actions can be grouped by user, IP address, /16 IP range, creation date of the user account, page, site, the edit count of the user or a combination thereof. (A simple example for throttling is something like "do this if page X is edited more than Y times in Z seconds".)

- *warn*: A warning is displayed that the edit may not be appreciated. (The warning message is configurable by the edit filter manager.) The editor who tripped the filter is provided with the opportunity to revise their edit and re-submit it. A link to the false positives page [Wik19aa] is also provided.

- *blockautopromote*: The user whose action matched the filter's pattern is banned from receiving extra groups from *$wgAutopromote* for a random period of 3 to 7 days.

- *block*: The user who triggered the filter is blocked indefinitely. An error message is displayed to inform the user of this action.

- *degroup*: The user whose action matched the filter's pattern is removed from all privileged groups (sysop, bureaucrat, etc). An error message is displayed to inform them of this action.

[6] See line 181 in https://gerrit.wikimedia.org/r/plugins/gitiles/mediawiki/extensions/AbuseFilter/+/refs/heads/master/includes/special/SpecialAbuseLog.php

[7] See line 2808 in https://gerrit.wikimedia.org/r/plugins/gitiles/mediawiki/extensions/AbuseFilter/+/refs/heads/master/includes/AbuseFilter.php

- *rangeblock*: The entire /16 IP range from which the filter was triggered is blocked for a week.

- *disallow*: An error message is shown to the editor informing them their edit was considered unconstructive and will not be saved. They are provided the opportunity to report a false positive.

rangeblock, *block*, *degroup* have never been used on the EN Wikipedia, at least according to the logs. Those severer actions were discussed controversially by the community before introducing the extension and a lot of Wikipedians felt uncomfortable with a fully automated mechanism blocking users indefinitely or removing them from privileged groups [Wik19ac], see also section 4.4. As far as I can tell, the functionality has been implemented but never activated (at least on the EN Wikipedia). The last time filter actions other than *log*, *tag*, *warn* or *disallow* were triggered on the EN Wikipedia was in 2012 and these were *blockautopromote* and *aftv5flagabuse* [8].

Guidelines specifically call for careful use of *disallow*. Only severe cases for which "substantially all good-faith editors would agree are undesirable" or specific cases for which consensus has been reached should be disallowed [Wik19r].

Following new user permissions are introduced by the AbuseFilter plugin:

- *abusefilter-modify*: "Modify abuse filters"

- *abusefilter-view*: "View abuse filters"

- *abusefilter-log*: "View the abuse log"

- *abusefilter-log-detail*: "View detailed abuse log entries"

- *abusefilter-private*: "View private data in the abuse log"

- *abusefilter-modify-restricted*: "Modify abuse filters with restricted actions"

- *abusefilter-modify-global*: "Create or modify global abuse filters"

- *abusefilter-revert*: "Revert all changes by a given abuse filter"

- *abusefilter-view-private*: "View abuse filters marked as private"

- *abusefilter-log-private*: "View log entries of abuse filters marked as private"

- *abusefilter-hide-log*: "Hide entries in the abuse log"

[8] *aftv5flagabuse* is a deprecated action related to the now deprecated Article Feedback MediaWiki extension (or Article Feedback Tool, Version 5) whose purpose was to involve readers more actively in article quality assessment [Wik19i]. However, during the testing phase the majority of reader feedback was found not particularly helpful and hence the extension was discontinued.

- *abusefilter-hidden-log*: “View hidden abuse log entries”
- *abusefilter-private-log*: “View the AbuseFilter private details access log”

For additional reference, the format for the rules [Med19d], the general documentation of the extension [Med19c], as well as its source code [Gar19a] can be consulted.

4.4. History

Now that there is a general understanding of what edit filters look like today, let us take a step back and investigate how they came to be this way. In order to comprehend the consensus building on the functionality of the extension, I sifted through the archives of the Edit Filter talk page [Wik19ac] for the period between the announcement that the extension is planned up until the voting process preceding its introduction.

For a while at the beginnings of the discussion, there was some confusion among editors regarding the intended functionality of the edit filters. Participants invoked various motivations for the introduction of the extension (which sometimes contradicted each other) and argued for or against the filters depending on these. The discussion reflects a mix of ideological and practical concerns. The biggest controversies lay along the lines of filters being public vs. private (hidden from public view) [9] and the actions the filters were to invoke upon a match. An automated rights revocation or a block of the offending editor with no manual confirmation by a real person were of particular concern to a lot of editors (they were worried that the filters would not be able to understand context thus resulting in too many false positives and blocking many legitimate edits and editors). As far as I understood, these features were technically implemented but never really used on English Wikipiedia.

As to the public vs. private debate, the initial plan was that all filters are hidden from public view and only editors with special permissions (the edit filter managers) were supposed to be able to view and modify the patterns and consult the logs. The core developer of the extension was reasoning that its primary purpose was to fend off really persistent vandals with reasonable technical understanding who were ready to invest time and effort to circumvent anti-vandal measures and that it was therefore unwise to make circumvention easier to them by allowing them to view the pattern according to which their edits were suppressed. This was however met with serious resistance by the community who felt that such secret extension was contradicting Wikipedia’s values of openness and transparency. Besides, opponents of the filters being completely private were concerned that the tool was quite powerful and hiding everything will prevent the community from monitoring for errors and abuse.

[9] The terms “private” and “hidden” are used interchangeably for such filters throughout the thesis.

Related to the above discussion, there was also disagreement regarding who was to have access to the newly developed tool. Some felt access had to be granted more broadly in order for the tool to be effectively used. They were reasoning that at least all administrators should be able to use it, since they already had the trust of the community. Others feared that the admin group was quite broad already and access should be granted as carefully as possible since the extension had the potential to cause quite a lot of damage if used maliciously and it was "not beyond some of our more dedicated trolls to "work accounts up" to admins, and then use them for their own purpose" [Wik19ac]. This narrower option is how the right ended up to be governed [10].

Another debated point was what the difference to bots (with admin rights) was and whether the extension was needed at all. Apparently, there was some discontent with bot governance mirrored in the arguments for introducing the extension. It was underlined that in contrast to admin bots the extension's source code was to be publicly available and well tested with more people (i.e. the edit filter managers) using and monitoring its functionality than the (usually) single bot operator responsible for a bot who, apparently, was oftentimes not responding to community concerns and emergencies fast enough (or at all). On the other hand, there were yet again arguments, that the extension was supposed to indeed target the really malicious vandals not deterred by anti-vandalism measures already in force by preferably blocking them on the spot.

Others were asking what additional advantages the extension offered compared to semi-protection of pages which requires users to be autoconfirmed in order to be able to edit (normally meaning they have to have been registered for at least 4 days and have made at least 10 edits, but the restrictions are adjustable). Here, User:Werdna was reasoning that the Edit Filters allow for fine-tuning of such restrictions and targeting offending editors in particular without making constraints unnecessarily strict for all users.

Although there were some diverging opinions on what the extension was supposed to target, in a nutshell, the motivation for its introduction seems to have been as follows: Bots weren't reverting some kinds of vandalism fast enough, or, respectively, these vandalism edits required a human intervention and took more than a single click to get reverted. These were mostly obvious but pervasive cases of vandalism (e.g. moving a lot of pages to some nonsensical name), possibly introduced in a (semi-)automated fashion, that took some time and effort to clean up. The motivation of the extension's developers was that if a filter just disallows such vandalism, vandal fighters could use their time more productively and check less obvious cases for which more background knowledge/context is needed in order to decide whether an edit is vandalism or not. According to the discussion archives, following types of edits were supposed to be targeted by the extension:
https://en.wikipedia.org/wiki/Special:Contributions/Omm_nom_nom_nom

[10] Although motivated trolls can potentially work up an account to any user rights.

https://en.wikipedia.org/wiki/Special:Contributions/AV-THE-3RD
https://en.wikipedia.org/wiki/Special:Contributions/Fuzzmetlacker

4.5. Building a Filter: the Internal Perspective

4.5.1. How Is a New Filter Introduced?

Only edit filter managers have the permissions necessary to implement filters, but anybody can propose new ones. Every editor who notices some problematic behaviour they deem needs a filter can raise the issue at the Edit Filter Requested page [Wik19ay]. The request can then be approved and implemented by an edit filter manager (mostly after a discussion/clarification of the details). The Edit Filter Requested page asks users to go through the following checklist before requesting a filter:

- problems with a single page are not suitable for an edit filter, since filters are applied to all edits;
- filters, after adding up, make editing slower, so the usefulness of every single filter and condition has to be carefully considered;
- in depth checks should be done by a separate software that users run on their own machines;
- no trivial errors should be caught by filters (e.g. concerning style guidelines);
- there are the Titles Blacklist [Med19g] and the Link/Spam Blacklist [Med19f] which should be used if the issue at hand has to do with a problematic title or link.

For edit filter managers, the best practice way for introducing a new filter is described on the Edit Filter Instructions page [Wik19al]. According to it, these steps should be followed:

1. read the documentation [Med19d],
2. test with debugging tools: https://en.wikipedia.org/wiki/Special:AbuseFilter/tools (visible only for users who are already in the edit filter managers user group),
3. test with the batch testing interface (also available to edit filter managers only),
4. create a logging only filter,
5. announce the filter at the edit filter notice board [Wik19y], so other edit filter managers can comment on it,

6. finally, fully enable the filter by adding an appropriate additional edit filter action.

According to the documentation, step 4 from the checklist can be skipped in "urgent situations" and corresponding filters can have severer actions enabled directly. In such case, the editor introducing the filter has to closely monitor the logs for potential filter misconduct. However, the guidelines do not elaborate on what exactly constitutes a "urgent situation". Unfortunately, investigating potential pitfalls of this provision is beyond the scope of the present work and one of the directions for further studies suggested in section 6.6.

Edit filter managers often introduce filters based on some phenomena they have observed caught by other filters, other algorithmic quality control mechanisms or general experience. As all newly implemented filters, these are initially enabled in logging only mode until enough log entries are generated to evaluate whether the incident is severe and frequent enough to need a filter.

It is not uncommon, that the action(s) a particular filter triggers change over time. Sometimes, when a wave of particularly persistent vandalism arises, a filter is temporarily set to "warn" or "disallow" and the actions are removed again as soon as the filter is not tripped very frequently anymore. Such action changes, updates to an edit filter's pattern, or a warning template, as well as problems with filters behaviour are discussed on the Edit Filter Noticeboard [Wik19y].

Last but not least, performance seems to be fairly important for the edit filter system: On multiple occasions, there are notes on recommended order of operations, so that the filter evaluates as resource sparing as possible [Wik19al] or invitations to consider whether an edit filter is the most suitable mechanism for solving a particular issue at all [Wik19r], [Wik19ay]. To optimise performance, the edit filter system uses the so-called condition limit. According to the documentation [Wik19u], the condition limit is a hard-coded treshold of total available conditions that can be evaluated by all active filters per incoming edit. Currently, it is set to $1,000$. The motivation for this heuristic is to avoid performance issues since every incoming edit is checked against all currently enabled filters which means that the more filters are active the longer the checks take. However, the page also warns that counting conditions is not the ideal metric of filter performance, since there are simple comparisons that take significantly less time than a check against the *all_links* variable for example (which needs to query the database) [Wik19u]. Nevertheless, the condition limit seems to still be the heuristic used for filter performance optimisation today.

4.5.2. Who Can Edit Filters?

In order to be able to set up an edit filter on their own, an editor needs to have the *abusefilter-modify* permission (which makes them part of the edit filter manager group). According to [Wik19r] this right is given only to editors who "have the required good judgment and technical proficiency". Further down on the page it is clarified that it is administrators who can assign the permission to users (also to themselves) and they should only assign it to non-admins in exceptional cases, "to highly trusted users, when there is a clear and demonstrated need for it". If editors wish to be given this permission, they can hone and prove their skills by helping with requested edit filters and false positives [Wik19r].

The formal process for requesting the *abusefilter-modify* permission is to raise the request at the Edit Filter Noticeboard [Wik19y]. A discussion is held there, usually for 7 days, before a decision is reached [Wik19r] [11].

As of 2017, when the "edit filter helper" group was introduced (editors in this group have the *abusefilter-view-private* permission) [Wik19w], the usual process seems to be that editors request this right first and only later the full *abusefilter-modify* permissions[12].

According to the edit filter managers list for the EN Wikipedia [Wik19x], as of 10 May 2019 there are 154 users in this group [13]. Out of the 154 edit filter managers only 11 are not administrators (most of them have other privileged groups such as "rollbacker", "pending changes reviewer", "extended confirmed user" and similar though).

The edit filter managers group is quite stable, with only 4 users who have become an edit filter manager since November 2016 (according to the archives of the edit filter noticeboard where the permission is requested) [Wik19y]. Since the edit filter helper group has been created in September 2017, only 11 users have been granted the corresponding permissions and only one of them has been subsequently "promoted" to become an edit filter manager [14].

Moreover, a number of the 154 edit filter managers on English Wikipedia have a kind of "not active at the moment" banner on their user page, which leads to the conclusion that the edit filter managers group is aging.

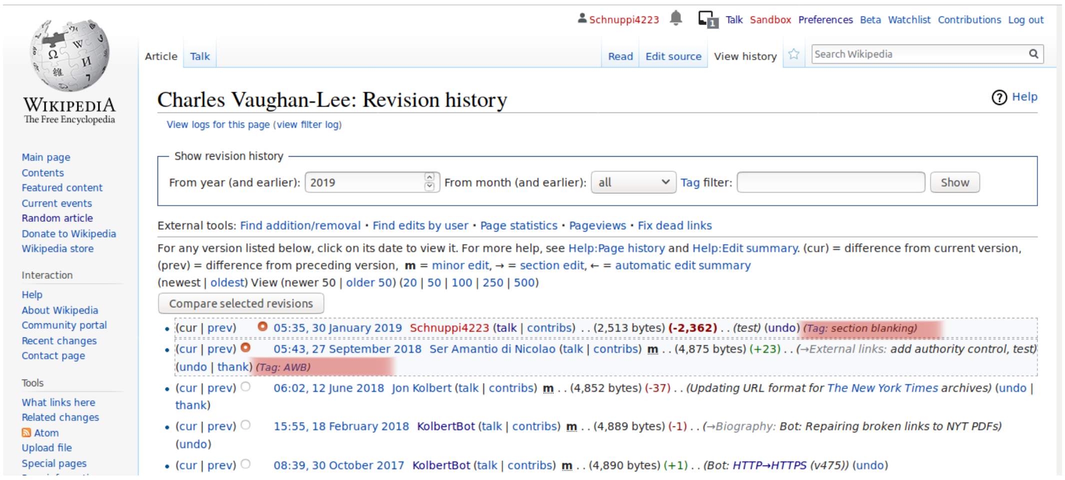
Some of the edit filter managers are also bot operators. The interesting patterns of collaboration between the two technologies are discussed in section 4.7.2.

[11] According to the documentation, the Edit Filter Noticeboard is also the place to discuss potential permission withdraws in cases of misuse where raising the issue directly with the editor concerned has not resolved the problem.

[12] That is the tendency observed at the Edit filter noticeboard [Wik19y].

[13] For comparison, as of 9 March 2019 there are 1181 admins [Wik19am]. The role does not exist at all on the German, Spanish and Russian Wikipedias where all administrators have the *abusefilter_modify* permission [Wik19t], [Wik19v], [Wik19ab].

[14] Interestingly, as of July 2019 there are 19 people in the edit filter helpers group, so apparently some of them have received the right although no records could be found on the noticeboard.



Charles Vaughan-Lee: Revision history

View logs for this page (view filter log)

Show revision history

From year (and earlier): 2019 From month (and earlier): all Tag filter: Show

External tools: Find addition/removal • Find edits by user • Page statistics • Pageviews • Fix dead links

For any version listed below, click on its date to view it. For more help, see Help:Page history and Help:Edit summary. (cur) = difference from current version, (prev) = difference from preceding version, **m** = minor edit, → = section edit, ← = automatic edit summary

(newest | oldest) View (newer 50 | older 50) (20 | 50 | 100 | 250 | 500)

Compare selected revisions

- (cur | prev) ● 05:35, 30 January 2019 Schnuppi4223 (talk | contribs) . . (2,513 bytes) **(-2,362)** . . *(test)* (undo) *(Tag: section blanking)*
- (cur | prev) ● 05:43, 27 September 2018 Ser Amantio di Nicolao (talk | contribs) **m** . . (4,875 bytes) (+23) . . *(→External links: add authority control, test)* (undo | thank) *(Tag: AWB)*
- (cur | prev) ○ 06:02, 12 June 2018 Jon Kolbert (talk | contribs) **m** . . (4,852 bytes) (-37) . . *(Updating URL format for The New York Times archives)* (undo | thank)
- (cur | prev) ○ 15:55, 18 February 2018 KolbertBot (talk | contribs) **m** . . (4,889 bytes) (-1) . . *(→Biography: Bot: Repairing broken links to NYT PDFs)* (undo)
- (cur | prev) ○ 08:39, 30 October 2017 KolbertBot (talk | contribs) **m** . . (4,890 bytes) (+1) . . *(Bot: HTTP→HTTPS (v475))* (undo)

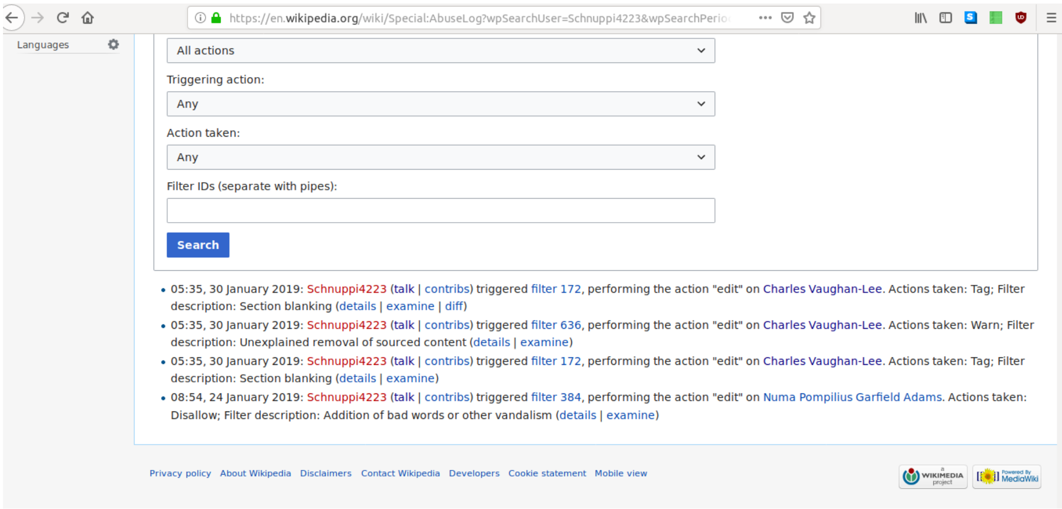
Figure 4.2.: Tagged edits are marked as such in a page's revision history

4.6. Filters during Runtime: the External Perspective

So what happens when an editor's action matches the pattern of an edit filter? Do they notice this at all?

As described section 4.3, a variety of different actions may occur when a filter's pattern matches. Of these, only *tag*, *throttle*, *warn*, and *disallow* seem to be used today (and *log*, which is always enabled). If a filter is set to *warn* or *disallow*, the editor is notified that they hit a filter by a warning message (see figure 4.4). These warnings describe the problem that occurred and present the editor with possible paths of action: complain on the False Positives page [Wik19aa] in case of *disallow* (the edit is not saved), or, complain on the False Positives page [15] and publish the change anyway in case of *warn*. (Of course, in case of a warning, the editor can modify their edit before publishing it.) Possible alternative paths of action an editor may wish to consult are also listed. On the other hand, when the filter action is set to *tag* or *log* only, the editor doesn't really notice they tripped a filter unless they are looking more closely. Tagged edits are marked as such in the page's revision history for example (see figure 4.2) and all edits that trigger an edit filter are listed in the Abuse Log [Wik19e] (see figure 4.3).

[15] Edit filter managers and other interested editors monitor the False Positives page and verify or disprove the reported incidents. Edit filter managers use actual false positives to improve the filters, give advice to good faith editors who tripped a filter and discourage authors of vandalism edits who reported these as false positives from continuing with their disruption.



All actions

Triggering action: Any

Action taken: Any

Filter IDs (separate with pipes):

- 05:35, 30 January 2019: Schnuppi4223 (talk | contribs) triggered filter 172, performing the action "edit" on Charles Vaughan-Lee. Actions taken: Tag; Filter description: Section blanking (details | examine | diff)
- 05:35, 30 January 2019: Schnuppi4223 (talk | contribs) triggered filter 636, performing the action "edit" on Charles Vaughan-Lee. Actions taken: Warn; Filter description: Unexplained removal of sourced content (details | examine)
- 05:35, 30 January 2019: Schnuppi4223 (talk | contribs) triggered filter 172, performing the action "edit" on Charles Vaughan-Lee. Actions taken: Tag; Filter description: Section blanking (details | examine)
- 08:54, 24 January 2019: Schnuppi4223 (talk | contribs) triggered filter 384, performing the action "edit" on Numa Pompilius Garfield Adams. Actions taken: Disallow; Filter description: Addition of bad words or other vandalism (details | examine)

Figure 4.3.: Abuse Log showing all filter matches by User:Schnuppi4223

4.7. Edit Filters' Role in the Quality Control Ecosystem

The purpose of the present section is to review what has been learnt so far about edit filters and summarise how they fit in Wikipedia's quality control ecosystem.

As timeline 4.1 shows, the time span in which algorithmic quality control mechanisms (first vandal fighting bots and semi-automated tools, and later filters) were introduced fits logically the period after the exponential growth of Wikipedia took off in 2006 (compare figures 4.5, 4.6). The surge in numbers of editors and contributions implied a rapidly increasing workload for community members dedicated to quality assurance which could not be feasibly handled manually anymore and thus the community turned to technical solutions. As shown elsewhere [HGMR13], this shift had a lot of repercussions—one of the most severe of them being that newcomers' edits were reverted stricter than before (accepted or rejected on a yes-no basis with the help of automated tools, instead of manually seeking to improve the contributions and "massage" them into an acceptable form), which in consequence drove a lot of them away.

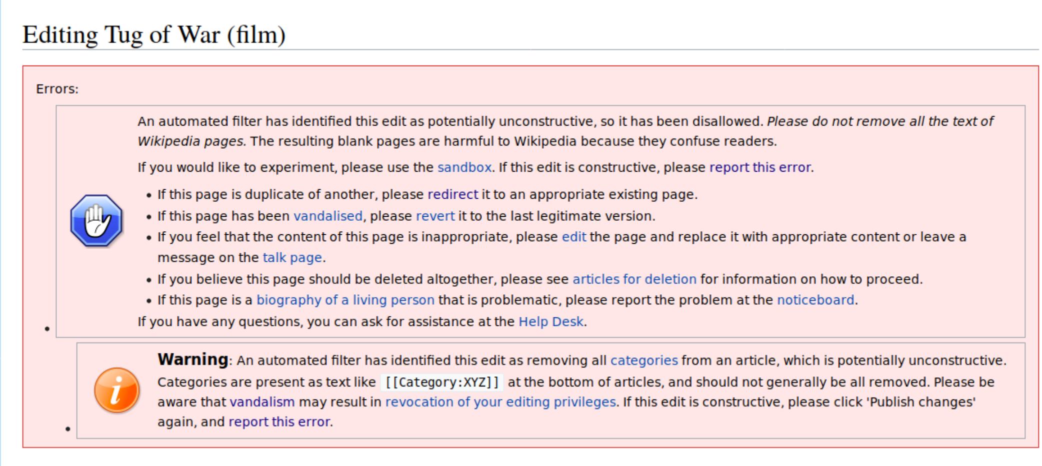
4.7.1. Wikipedia's Algorithmic Quality Control Mechanisms in Comparison

As we can read from timeline 4.1, edit filters were introduced at a moment when bots and semi-automated tools were already in place. Thus, the question arises: Why were they implemented when already these other mechanisms existed? Here, the salient features of the different quality control mechanisms and the motivation for the filters' introduction are reviewed. A concise summary of this discussion is offered in table 4.2.

Since edit filters are a fully automated mechanism, above all a comparison

Oct 2001	automatically import entries from Easton's Bible Dictionary by a script
29 Mar 2002	First version of https://en.wikipedia.org/wiki/Wikipedia:Vandalism (WP Vandalism is published)
Oct 2002	RamBot
2006	The Bot Approvals Group (BAG) was first formed
13 Mar 2006	1st version of Bots/Requests for approval is published: some basic requirements (also valid today) are recorded
28 Jul 2006	VoABot II (”In the case were banned users continue to use sock-puppet accounts/IPs to add edits clearly rejected by consensus to the point were long term protection is required, VoABot may be programmed to watch those pages and revert those edits instead. Such edits are considered blacklisted. IP ranges can also be blacklisted. This is reserved only for special cases.”)
21 Jan 2007	Twinkle Page is first published (empty), filled with a basic description by beginnings of Feb 2007
24 Jul 2007	Request for Approval of original ClueBot
16 Jan 2008	Huggle Page is first published (empty)
18 Jan 2008	Huggle Page is first filled with content
23 Jun 2008	1st version of Edit Filter page is published: User:Werdna announces they're currently developing the extension
2 Oct 2008	https://en.wikipedia.org/wiki/Wikipedia_talk:Edit_filter was first archived; its last topic was the voting for/against the extension which seemed to have ended end of Sep 2008
Mar 2009	The AbuseFilter extension is enabled on English Wikipedia
Jun 2010	STiki initial release
20 Oct 2010	ClueBot NG page is created
11 Jan 2015	1st commit to github ORES repository
30 Nov 2015	ORES paper is published

Table 4.1.: Timeline: Introduction of algorithmic quality control mechanisms



Editing Tug of War (film)

Errors:

An automated filter has identified this edit as potentially unconstructive, so it has been disallowed. *Please do not remove all the text of Wikipedia pages.* The resulting blank pages are harmful to Wikipedia because they confuse readers.

If you would like to experiment, please use the sandbox. If this edit is constructive, please report this error.

- If this page is duplicate of another, please redirect it to an appropriate existing page.
- If this page has been vandalised, please revert it to the last legitimate version.
- If you feel that the content of this page is inappropriate, please edit the page and replace it with appropriate content or leave a message on the talk page.
- If you believe this page should be deleted altogether, please see articles for deletion for information on how to proceed.
- If this page is a biography of a living person that is problematic, please report the problem at the noticeboard.

If you have any questions, you can ask for assistance at the Help Desk.

Warning: An automated filter has identified this edit as removing all categories from an article, which is potentially unconstructive. Categories are present as text like `[[Category:XYZ]]` at the bottom of articles, and should not generally be all removed. Please be aware that vandalism may result in revocation of your editing privileges. If this edit is constructive, please click 'Publish changes' again, and report this error.

Figure 4.4.: Editor gets two warnings upon erasing an entire page: one for page blanking and another one for removing all categories from an article. The warnings list possible pages the editor may want to consult and actions they can take.

to bots seems obvious. The main argument for introducing the extension were the usecases it was supposed to take care of: the obvious persistent vandalism (often automated itself) which was easy to recognise but more difficult to clean up. Filters were going to do the job more neatly than bots by reacting faster, since the extension was part of the core software, and since they are triggered *before* an edit is published–by not allowing abusive content to become public at all. By being able to disallow such malicious edits from the beginning, the extension was to reduce the workload of other mechanisms and free up resources for vandal fighters using semi-automated tools or monitoring pages manually to work on less obvious cases that required human judgement, reasoned proponents of the filters.

The rest of the arguments for edit filters vs. bots touched on in the discussion prior to introducing filter [Wik19ac] were more of infrastructural/soft nature. The plugin's developers optimistically announced that it was going to be open source, the code well tested, with framework for testing single filters before enabling them and edit filter managers being able to collaboratively develop and improve filters. They viewed this as an improvement compared to (admin) bots which would be able to cover similar cases but whose code was mostly private, not tested at all, and with a single developer/operator taking care of them who was often not particularly responsive in emergency cases [16].

[16] For the sake of completeness, it should be mentioned here that the most popular semi-automated anti-vandalism tools are also open sourced. Their focus however lies somewhat differently, since a final human decision is required, and that is why probably they are not mentioned at all in this discussion. ORES is open source as well, it is kind of a meta tool that can be employed by the other mechanisms though and that is a why a direct comparison is also not completely feasible. Besides, it was introduced some 6-7 years after the edit filters,

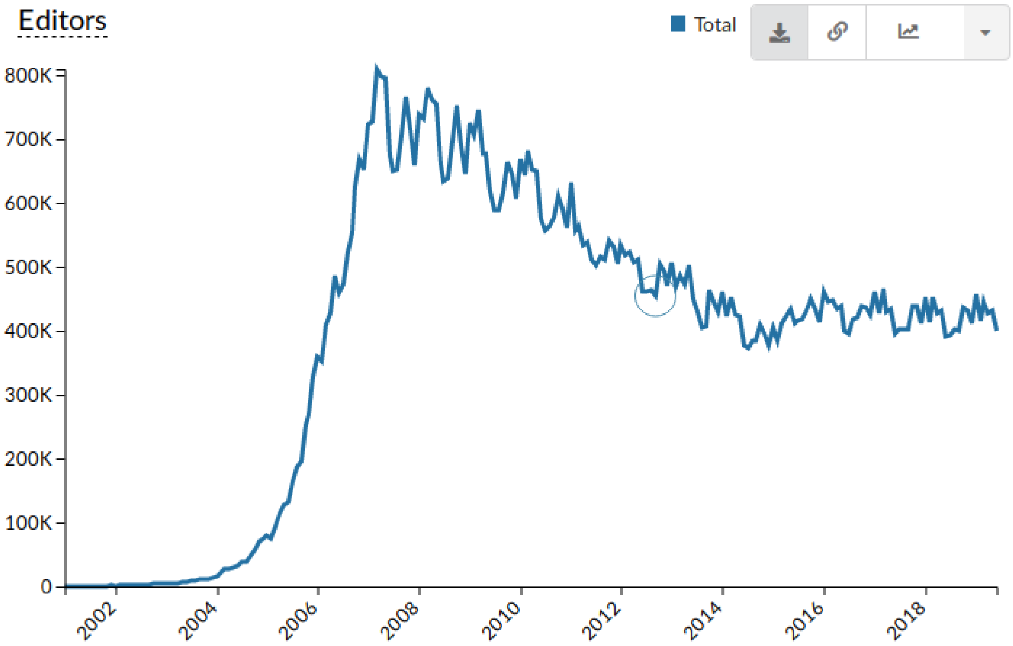


Figure 4.5.: EN Wikipedia: Number of editors over the years (source: https://stats.wikimedia.org/v2/)

Another apparent comparison is the one between edit filters and MediaWiki's page protection mechanism [Med19e]. As pointed out in section 2.1.3, page protection is reasonable when a rise in disruptive activity on a particular page occurs. Similarly to applying an edit filter aiming at the specific page, page protection would simply disallow edits to it from the start. The difference however is that edit filters could target a specific malicious user (or users) directly, without imposing restrictions on the vast majority of editors.

From all the mechanisms, it is probably the hardest to become engaged with edit filters. As signaled in section 4.5.2, the permissions are only granted to very carefully selected editors who have long history of participation on Wikipedia and mostly also various other special permissions. The numbers also demonstrate that this is the most exclusive group: as mentioned in section 4.5.2, there are currently 154 edit filter managers on EN Wikipedia, compared to at least 232 bot operators [Wik19l] (most likely not all bot operators are listed in the category [Wik19ah]) and 6130 users who have the *rollback* permission [Wik19az]. As to the difficulty/competences needed, it is probably easiest to learn to use semi-automated tools where one "only" has to master the user interface of the software. Bots require presumably most background knowledge since one has to not only be familiar with a programming language

so obviously people were not discussing it at the time.

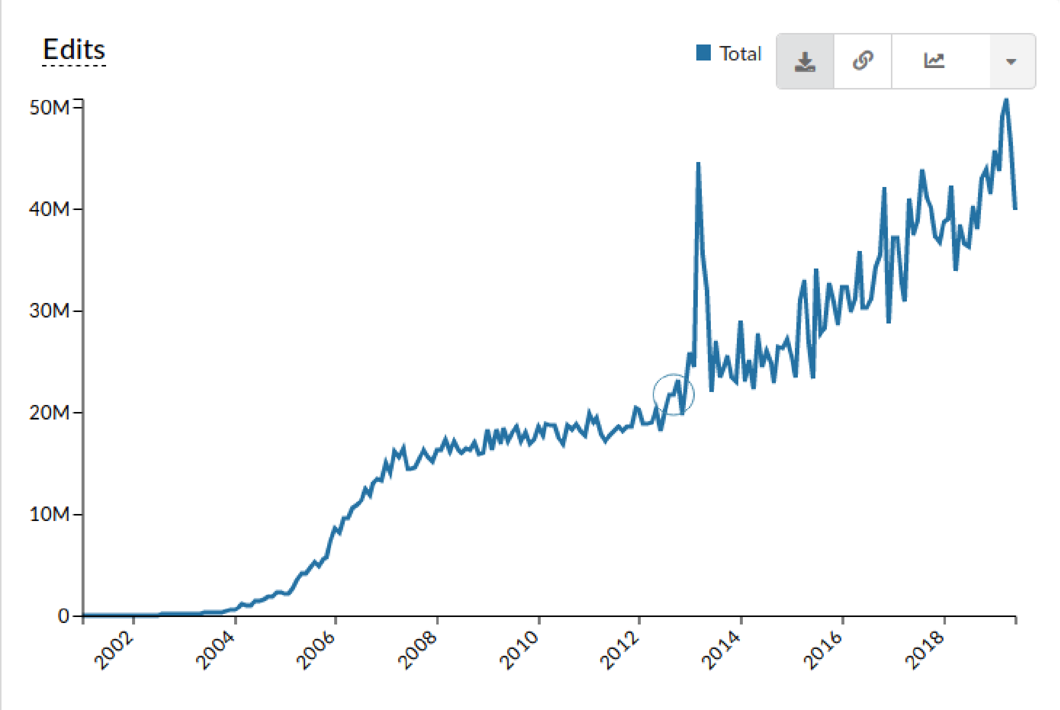


Figure 4.6.: EN Wikipedia: Number of edits over the years (source: https://stats.wikimedia.org/v2/)

but also learn to interact with Wikipedia's API, etc. Filters on the other hand, are arguably easier to use: here, "only" understanding of regular expressions is required.

As already summarised in chapter 2, critical voices worry about various aspects of the individual quality control mechanisms (see also table 4.2). Concerns with filters resemble somewhat the concerns expressed about bots: namely, the apprehension of a fully-automated mechanism taking (potentially erroneous) decisions about excluding editors from participation. In consequence, community consensus on using filter actions such as *rangeblock*, *block*, and *degroup* never happened. According to the discussion archives [Wik19ac], others feared that edit filters were placing a lot of power in the hands of very few people.

	Filters	Page protection	Blacklists	Bots	Semi-Automated tools	ORES
Properties	rule-based	per page	rule-based	rule/ML based	rule/ML based	ML framework
	part of the ”software” (MediaWiki plugin)	part of MediaWiki	part of MediaWiki	run on user’s infrastructure (”bespoke code”)	extra infrastructure	not used directly, can be incorporated in other tools
	open source	MediaWiki is open source		no requirement for code to be public	most popular are open source (but not a hard requirement)	open source
	public filters directly visible for anyone interested				heuristics obfuscated by the interface	
	trigger *before* an edit is published	one cannot edit at all	trigger before an edit is published	latency varies (ClueBot NG needs only a couple of seconds)	generally higher latency than bots	

	collaborative effort			mostly single dev/operator (recently: bot frameworks)	few devs	few devs
Necessary perminssions	*abusefilter-modify* permission	admin access	admin access bot gets a “bot flag”	no special rights needed (except for admin bots)	*rollback* permission needed	
People involved	edit filter managers (EN Wiki)	admins	admins (anyone can report problems)	bot operators	editors with *rollback* permission	mostly Scoring Platform team
Hurdles to participate	gain community trust to become an edit filter manager understand regexes		understand regexes	get approval from the BAG programming knowledge, understand APIs, ...	get a *rollback* permission get familiar with the tool	understand ML
Concerns	automated agents blocking/desysoping human users			“botophobia”	gamification	general ML concerns: hard to understand

	hidden filters lack transparency and accountability censorship infrastructure				interface makes some paths of action easier than others	
Areas of application	persistent vandal with a known modus operandi and a history of circumventing prevention methods' demographic (obvious vandalism which takes time to clean up)	problems with single page	abusive titles/ link spam	mostly obvious vandalism	less obvious cases that require human judgement	

Table 4.2.: Wikipedia's algorithmic quality control mechanisms in comparison

4.7.2. Collaboration of the Mechanisms

So far, the single quality control mechanisms have been juxtaposed and separately compared. It is however worth mentioning that they not only operate alongside each other but also cooperate on occasions.

Such collaborations are studied for instance by Geiger and Ribes [GR10] who go as far as describing them as "distributed cognition". They follow a particular series of abuse throughout Wikipedia, along the traces the disrupting editor and the quality control mechanisms deployed against their edits left. The researchers demonstrate how a bot (ClueBot), and several editors using the semi-automated tools Huggle and Twinkle all collaborated up until the malicious editor was banned by an administrator.

During the present study, I have also observed various cases of edit filters and bots mutually facilitating each other's work. DatBot, Mr.Z-bot and MusikBot are all examples for bots conducting support tasks for filters. DatBot [Wik19p] monitors the Abuse Log [Wik19e] and reports users tripping certain filters to WP:AIV (Administrator intervention against vandalism)[Wik19f] and WP:UAA (usernames for administrator attention) [Wik19be]. It is the successor of Mr.Z-bot [Wik19aq] which used to report users from the abuse log to WP:AIV, but has been inactive since 2016 and therefore recently deactivated.

MusikBot also has several tasks dedicated to monitoring different aspects of edit filter behaviour and compiling reports for anyone interested: The FilterMonitor task "[r]eports functional changes of edit filters to the watchable page User:MusikBot/FilterMonitor/Recent changes. The template `{{recent filter changes}}` formats this information and can be transcluded where desired" [Wik19as]. The StaleFilter task "[r]eports enabled filters that have not had any hits in over 30 days, as specified by `/Offset`" [Wik19at]. The AbuseFilterIRC task "[r]elays all edit filter hits to IRC channels and allows you to subscribe to notifications when specific filters are tripped. See `#wikipedia-en-abuse-log-all` for the English Wikipedia feed" [Wik19ar].

On the other hand, there are also examples for filters supporting bot work: Filter 323 ("Undoing anti-vandalism bot") tags edits reverting revisions by XLinkBot and ClueBot NG. Although it is hidden, so no details can be viewed by a unauthorised user, filter 603 is named "Special case of reverting XLinkBot reverts" so it is probably safe to assume that is filtering what it claims to be. And there are several filters (historically) configured to ignore particular bots: filter 76 ("Adding email address") exempting XLinkBot, filter 28 ("New user redirecting an existing substantial page or changing a redirect") exempting Anybot, filter 532 ("Interwiki Addition") exempting Cydebot are some examples thereof. There are also filters configured to ignore all bots: filter 368 ("Making large changes when marking the edit as minor"), filter 702 ("Warning against clipboard hijacking"), filter 122("Changing Username malformed requests").

On occasions, data from the Abuse Log is used for (semi-)protecting frequently disrupted pages.

And as discussed in chapter 2, ORES scores can be employed by bots or semi-automated tools as a heuristic to detect potentially harmful edits. Note that edit filters cannot use ORES, since the service computes scores according to different models for already published revisions.

4.7.3. Conclusions

In short, this chapter studied edit filters' documentation and community discussions and worked out the salient characteristics of this mechanism. Moreover, the filters were compared to other quality control technologies on Wikipedia such as bots, semi-automated anti-vandalism tools and the machine learning framework ORES. Edit filters were considered in the context and time of their introduction and it was concluded that the community implemented them as a means to fight obvious, particularly persistent, and cumbersome to remove vandalism by disallowing it on the spot. Other "softer" arguments such as dissatisfaction with bot development processes (poorly tested, non-responsive operators) seemed to encourage the introduction as well. It was found that the individual filters are implemented and maintained by edit filter managers, a special highly-restricted user group.

Revising the quality control ecosystem diagram 2.1 introduced in chapter 2, filters can now be properly placed on it (see figure 4.7). It seems that claims of the literature (see section 2.1.1) should be revised: In terms of temporality not bots but edit filters, alongside page protection and blacklists, are the first mechanism to actively fend off a disruptive edit.

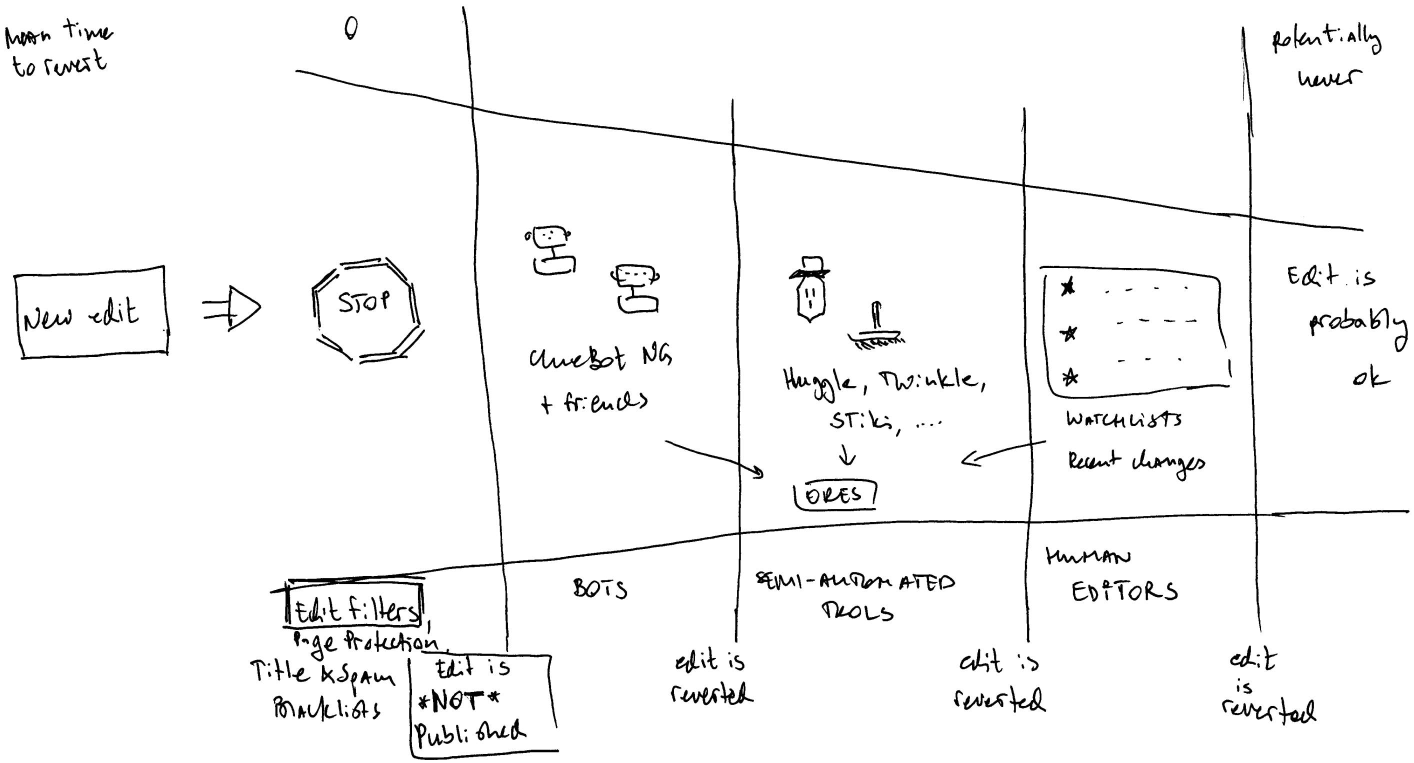


Figure 4.7.: Edit filters' role in the quality control ecosystem: They are the first mechanism to get active.

5. Descriptive Overview of Edit Filters on the English Wikipedia

After tracing the debate surrounding the introduction of edit filters and piecing together how they work and what their supposed purpose was in chapter 4, here I explore the edit filters currently existent on the English Wikipedia. I want to gather a understanding of what types of tasks these filters take over, in order to compare them to the declared aim of the filters, and, as far as feasible, trace how these tasks have evolved over time.

The data upon which the analysis is based is described in section 5.1 and the methods used—in chapter 3. The manual classification of EN Wikipedia's edit filters I've undertaken in an attempt to understand what is it that they actually filter is presented in section 5.2. Section 5.3 studies characteristics of the edit filters in general, whereas their activity is analysed in section 5.4.

5.1. Data

A big part of the present analysis is based upon the *abuse_filter* table from *enwiki_p* (the database which stores data for the EN Wikipedia), or more specifically a snapshot thereof which was downloaded on 6 January 2019 via quarry, a web-based service offered by Wikimedia for running SQL queries against their public databases [1]. The complete dataset can be found in the repository for the present paper [Git19]. This table, along with *abuse_filter_actions*, *abuse_filter_log*, and *abuse_filter_history*, are created and used by the AbuseFilter MediaWiki extension ([Gar19b]), as discussed in section 4.3.

Selected queries have been run via quarry against the *abuse_filter_log* table as well. These are the foundation for the filters activity analysis undertaken in section 5.4. Unfortunately, the *abuse_filter_history* table which will be necessary for a complete historical analysis of the edit filters is currently not exposed to the public due to security/privacy concerns [Pla16b] [2]. A comprehensive historical analysis is therefore one of the directions for future research discussed in section 6.6.

A concise description of the tables has been offered in section 4.3 which discusses the AbuseFilter MediaWiki extension in more detail. For further reference, the schemas of all four tables can be viewed in figures B.1, B.2, B.3 and B.4 in the appendix.

[1] https://quarry.wmflabs.org/

[2] A patch was submitted to Wikimedia's operations repository where the replication scripts for all publicly exposed databases are hosted [Mes19]. It is in a process of review, so hopefully, historical filter research will be possible in the future.

5.2. Types of Edit Filters: Manual Classification

In order to get a better understanding of what exactly it is that edit filters are filtering, I applied emergent coding (see section 3.2) to all filters, scrutinising their names, patterns, comments, and actions. Three big clusters of codes were identified, namely “vandalism”, “good faith”, and “maintenance”, as well as the auxiliary cluster “unknown”. These are discussed in more detail later in this section, but first the coding itself is presented.

5.2.1. Coding Process and Challenges

As already mentioned, I applied emergent coding on the dataset from the *abuse_filter* table and let the labels originate directly from the data. I looked through the data paying special attention to the name of the filters (“af_public_comments” field of the *abuse_filter* table), the comments (“af_comments”), the pattern constituting the filter (“af_pattern”), and the designated filter actions (“af_actions”).

The assigned codes emerged from the data: some of them being literal quotes of terms used in the description or comments of a filter, while others summarised the perceived filter functionality. In addition to that, for vandalism related labels, I used some of the vandalism types elaborated by the community in [Wik19bh]. However, this typology was regarded more as an inspiration instead of being adopted 1:1 since some of the types were quite general whereas more specific categories seemed to render more insights. For instance, I haven’t applied the “addition of text” category since it seemed more useful to have more specific labels such as “hoaxing” or “silly_vandalism” (check the code book in the appendix A for definitions). Moreover, I found some of the proposed types redundant. For example, “sneaky vandalism” seems to overlap partially with “hoaxing” and partially with “sockpuppetry”, and for some reason, “personal attacks” are listed twice.

Based on the emergent coding method described in section 3.2, I have labeled the dataset twice. I let potential labels emerge during the first round of coding. Then, I scrutinised them, merging labels that seemed redundant and letting the most descriptive code stay. At the same time, the codes were also sorted and unified into broader categories which seemed to relate the single labels to each other. Thereby, a code book with the conclusive codes was defined (see appendix A). Subsequently, I labeled the whole dataset again using the code book. Unfortunately, the validation steps proposed by the method could not be realised, since no second researcher was available for the labeling. This is one of the limitations discussed in section 6.5, and respectively something that can and should be remedied in future research.

Following challenges were encountered during the first round of labeling: There were some ambiguous cases which I either tagged with the code I deemed most appropriate and a question mark, or assigned all possible labels (or both). There were also cases for which I could not gather any insight relying on the name, comments and pattern, since the filters were hidden from public view and the name was not descriptive enough. However, upon some further reflection, I think it is safe to assume that all hidden filters target a form of (more or less grave) vandalism, since the guidelines suggest that filters should not be hidden in the first place unless dealing with cases of persistent and specific vandalism where it could be expected that the vandalising editors will actively look for the filter pattern in their attempts to circumvent the filter[Wik19r]. Therefore, during the second round of labeling I tagged all hidden filters for which there weren't any more specific clues (for example in the name of the filter) as "hidden_vandalism". And then again, there were also cases, not necessarily hidden, where I could not determine any suitable label, since I didn't understand the pattern, and/or none of the existing categories seemed to fit, and/or I couldn't think of an insightful new category to assign. During the first labeling, these were labeled "unknown", "unclear" or "not sure". For the second round, I have unified all of them under "unclear".

For a number of filters, it was particularly difficult to determine whether they were targeting vandalism or good faith edits. The only thing that would have distinguished between the two would have been the contributing editor's motivation, which no one could have known (but the editor in question themself). During the first labeling session, I tended to label such filters with "vandalism?, good_faith?". For the second labeling, I stuck to the "assume good faith" guideline [Wik19k] myself and only labeled as vandalism cases where good faith was definitely out of the question. One feature which guided me here was the filter action which represents the judgement of the edit filter manager(s). Since communication is crucial when assuming good faith, all ambiguous cases which have a less "grave" filter action such as "tag" or "warn" (which seeks to give feedback and thereby effect/influence a constructive contribution) have received a "good_faith" label. On the other hand, filters set to "disallow" were tagged as "vandalism" or a particular type thereof, since the filter action is a clear sign that at least the edit filter managers have decided that seeking a dialog with the offending editor is no longer an option.

For the second round of labeling, I tagged the whole dataset again using the compiled code book (see A) and assigned to every filter exactly one label—the one deemed most appropriate (although oftentimes alternative possibilities were listed as notes), without looking at the labels I assigned the first time around. I intended to compare the labels from both coding sessions and focus on more ambiguous cases, re-evaluating them using all available information (patterns, public comments, labels from both sessions, as well as any notes I made along the line). Unfortunately, time was scarce, so the analysis of the present section is based upon the second round of labeling. Comparing codes from both labeling sessions and refining the coding, or respectively have another person label the data should be done in the future.

The datasets developed during both labeling sessions are available in project's repository [Git19].

As signaled at the beginning of the section, following four parent categories of codes were identified: “vandalism”, “good faith”, “maintenance”, and “unknown”. The subsections that follow discuss the salient properties of each of them.

5.2.2. Vandalism

The vast majority of edit filters on EN Wikipedia could be said to target (different forms of) vandalism, i.e. maliciously intended disruptive editing (or other activity) [Wik19bg]. Some examples thereof are filters for juvenile types of vandalism (inserting swear or obscene words or nonsense sequences of characters into articles), for hoaxing (inserting obvious or less obvious false information in articles), for template vandalism (modifying a template in a disruptive way which is quite severe, since templates are displayed on various pages), or for spam (inserting links to promotional content, often not related to the content being edited). All codes belonging to the vandalism category together with a definition and examples can be consulted in the code book attached in the appendix A.

Some vandalism types seem to be severer than others (e.g. sock puppetry [3] or persistent long term vandals). It is mostly in these cases that the implemented filters are hidden. Labels referring to such types of vandalism form their own subcategory: “hardcore vandalism”. It should be mentioned at this point that I also classified “harassment” and “personal attacks” as “hardcore vandalism”, since these types of edits are highly harmful and often dealt with by hidden filters, although according to [Wik19bg] both behaviours are disruptive editing rather than vandalism and should generally be handled differently.

5.2.3. Good Faith

The second biggest category identified were filters targeting (mostly) disruptive but not necessarily made with bad intentions edits. The adopted name “good faith” is a term utilised by the Wikipedia community itself, most prominently in the guideline “assume good faith” [Wik19k]. Filters from this category are frequently aimed at unconstructive edits done by new editors, not familiar with syntax, norms, or guidelines which results in broken syntax, disregard of established processes (e.g. deleting something without running it through an Articles for Deletion process, etc.) or norms (e.g. copyright violations), or unencyclopedic edits (e.g. without sources/with improper sources; badly styled; or with a skewed point of view).

The focus of these filters lies in the communication with the disrupting editors: a lot of the filters issue warnings intending to guide the editors towards

[3] Sock puppetry denotes the creation and employment of several accounts for various purposes such as pushing a point of view, or circumventing bans. For more information, see the code book in the appendix A

ways of modifying their contribution to become a constructive one (compare with section 4.6).

Codes from this category often take into consideration the area the editor was intending to contribute to or respectively that they (presumably) unintentionally disrupted.

5.2.4. Maintenance

Some of the encountered edit filters on the EN Wikipedia were targeting neither vandalism nor good faith edits. Rather, they had their focus on (semi-)automated routine (clean up) tasks. These filters form the "maintenance" category. Some of them target for instance bugs like broken syntax caused by a faulty browser extension. Or there are such which simply track particular behaviours (such as mobile edits or edits made by unflagged bots) for various purposes.

The "maintenance" category differs conceptually from the "vandalism" and "good faith" ones in so far that the logic behind it isn't editors' intention, but rather "side"-occurrences that mostly went wrong.

I've also grouped here various test filters (used by individual editors or jointly used by all editors).

5.2.5. Unknown

This is an auxiliary category comprising the "unknown" and "misc" codes used to code all filters where the functionality stayed completely opaque for the observer, or, although it was comprehensible what the filter was doing, still no better fitting label emerged.

5.3. Filter Characteristics

This section explores some general features of the edit filters on English Wikipedia based on the data from the *abuse_filter* table. The scripts that generate the statistics discussed here, can be found in the jupyter notebook in the project's repository [Git19].

5.3.1. General Traits

As of 6 January 2019 there are 954 filters in the *abuse_filter* table. It should be noted, that if a filter gets deleted, merely a flag is set to indicate so, but no entries are removed from the database. So, the above mentioned 954 filters are all filters ever made up to this date. This doesn't mean that it never changed what the single filters are doing, since edit filter managers can freely modify filter patterns, so at some point a filter could be doing one thing and in the next moment it can be filtering a completely different phenomenon. There are cases of filters being "re-purposed" or modified to filter for example a more

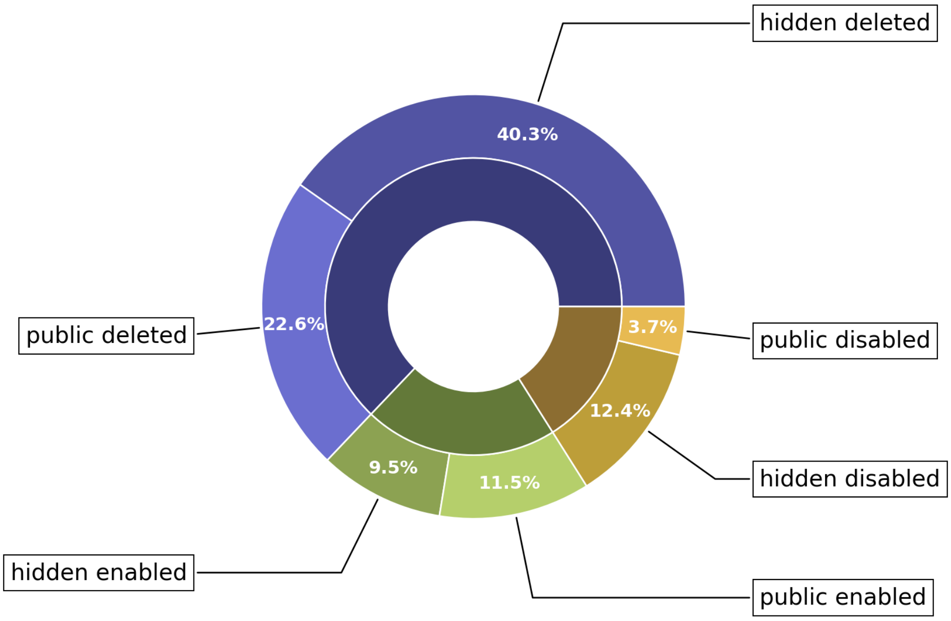


Figure 5.1.: There are 954 edit filters on EN Wikipedia: roughly 21% of them are active, 16% are disabled, and 63% are deleted

general occurrence. This doesn't happen very often though. Mostly, if a filter is not useful anymore, it is just disabled and eventually deleted and new filters are implemented for current problems.

361 of all filters are public, the remaining 593—hidden. 110 of the public ones are active, 35 are disabled, but not marked as deleted, and 216 are flagged as deleted. Out of the 593 hidden filters 91 are active, 118 are disabled (not deleted), and 384 are deleted. The relative proportion of these groups to each other can be viewed on figure 5.1.

5.3.2. Public and Hidden Filters

As signaled in section 4.4, historically it was planed to make all edit filters hidden from the general public. The community discussions rebutted that so a guideline was drafted calling for hiding filters "only where necessary, such as in long-term abuse cases where the targeted user(s) could review a public filter and use that knowledge to circumvent it." [Wik19r]. This is however not always complied with and edit filter managers do end up hiding filters that target general vandalism despite consensus that these should be public [Wik19ax]. Such cases are usually made public eventually (examples hereof are filters 225 "Vandalism in all caps", 260 "Common vandal phrases", or 12 "Replacing a page with obscenities"). Also, oftentimes when a hidden filter is marked as "deleted", it is made public.

Further, caution in filter naming is suggested for hidden filters and editors are encouraged to give such filters just simple description of the overall disruptive behaviour rather than naming a specific user that is causing the disruptions. (The latter is not always complied with, there are indeed filters named after the accounts causing a disruption.)

Still, it draws attention that currently nearly 2/3 of all edit filters are not viewable by the general public (compare figure 5.1). Unfortunately, without the full *abuse_filter_history* table there is no way to know how this ration has developed historically. However, the numbers fit the assertion of the extension's core developer according to whom edit filters target particularly determined vandals (filters aimed at whom are, as a general rule, hidden in order to make circumvention more difficult).

On the other hand, if we look at the enabled filters only, there are actually more or less the same number of public enabled and hidden enabled filters (110 vs. 91). This leads to the hypothesis that it is rather that hidden filters have higher fluctuation rates, i.e. that they target specific phenomena that are over after a particular period of time after which the filters get disabled and eventually—deleted. This again makes sense when compared to the hidden vs. public filter policy: hidden filters for particular cases and very determined vandals, public filters for general patterns which reflect more timeless patterns.

5.3.3. Filter Actions

Another interesting parameter observed here are the currently configured filter actions for each filter. Figure 5.2 depicts the actions set up for all enabled filters. And figures 5.3 and 5.4 show the actions of all enabled public and hidden filters respectively. It is noticeable that the most common action for the enabled hidden filters is "disallow" whereas most enabled public filters are set to "tag" or "tag,warn". This is congruent with the community's claim that hidden filters target particularly persistent vandalism, which is best outright disallowed. A lot of public filters on the other hand still assume good faith from the editors and try to dissuade them from engaging in disruptive behaviour by using warnings or just tag conspicuous behaviour for further investigation.

5.3.4. What Do Filters Target

This section examines in detail the results of the manual tagging of the filters according to their perceived functionality described in section 5.2. As figures 5.5 and 5.6 demonstrate, the majority of filters seem to target vandalism (little surprise here). The second biggest category comprise the "good faith" filters, while "maintenance" and "unknown" filters make up a relatively small part of the total number of filters. The proportion of vandalism related filters is higher when all filters are considered and not just the enabled ones. Again, this is probably due to the presumed higher fluctuation rates of hidden filters which (according to my labeling, see section 5.2 for rationale) are always vandalism related. It also comes to attention that the relative share of maintenance related filters is higher when all filters are regarded. The detailed

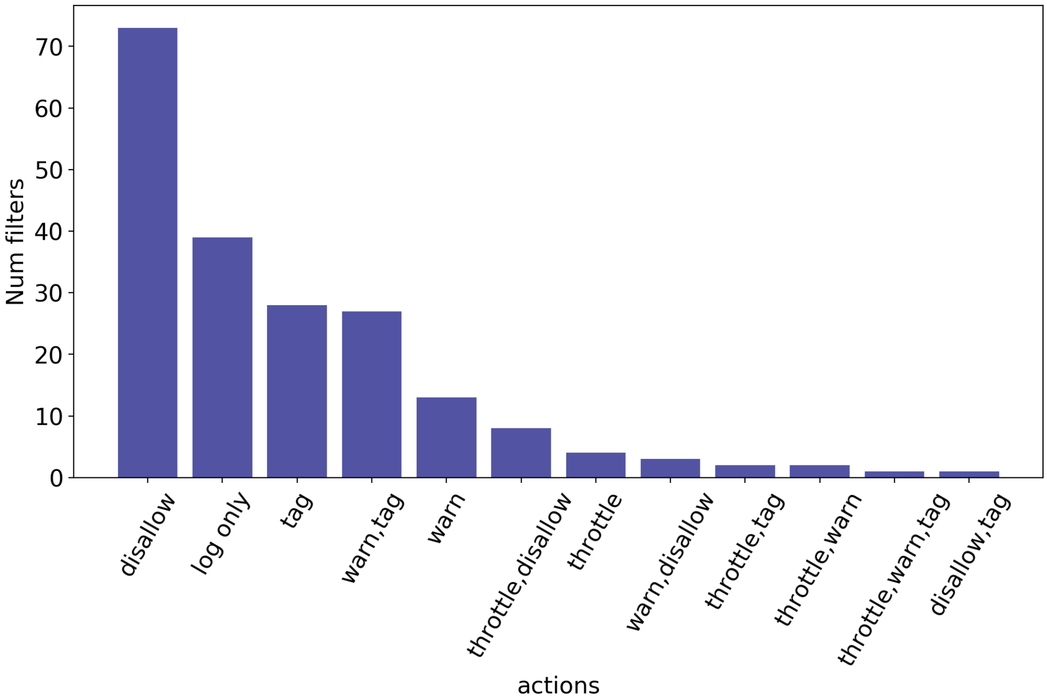


Figure 5.2.: EN Wikipedia edit filters: Filters actions for all filters

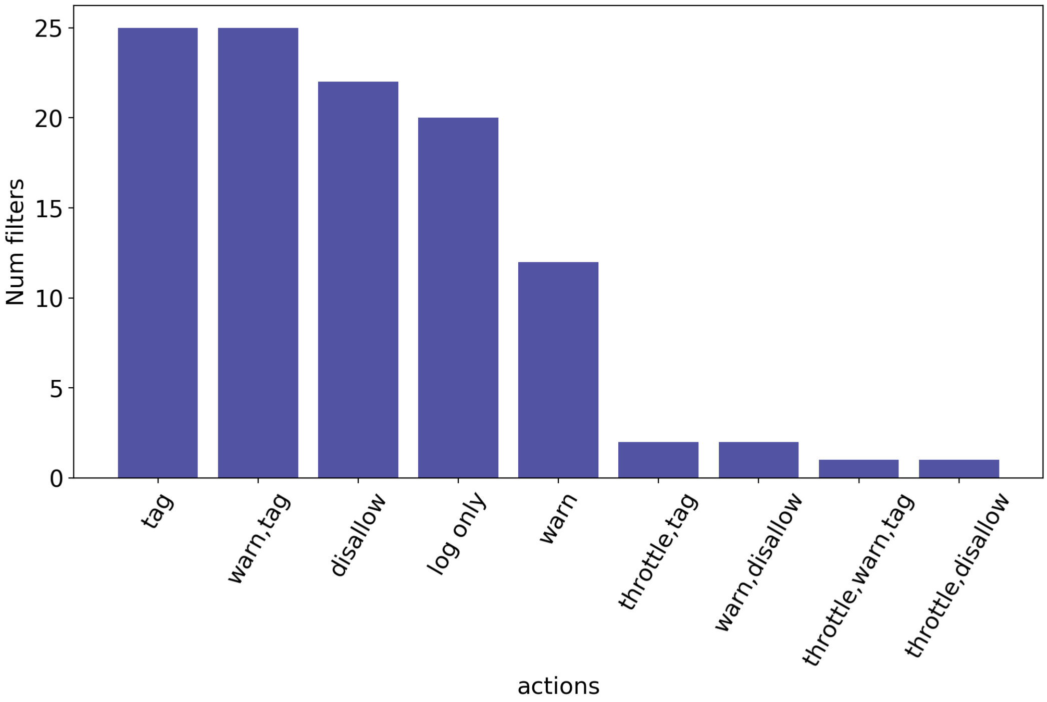


Figure 5.3.: EN Wikipedia edit filters: Filters actions for enabled public filters (January 2019)

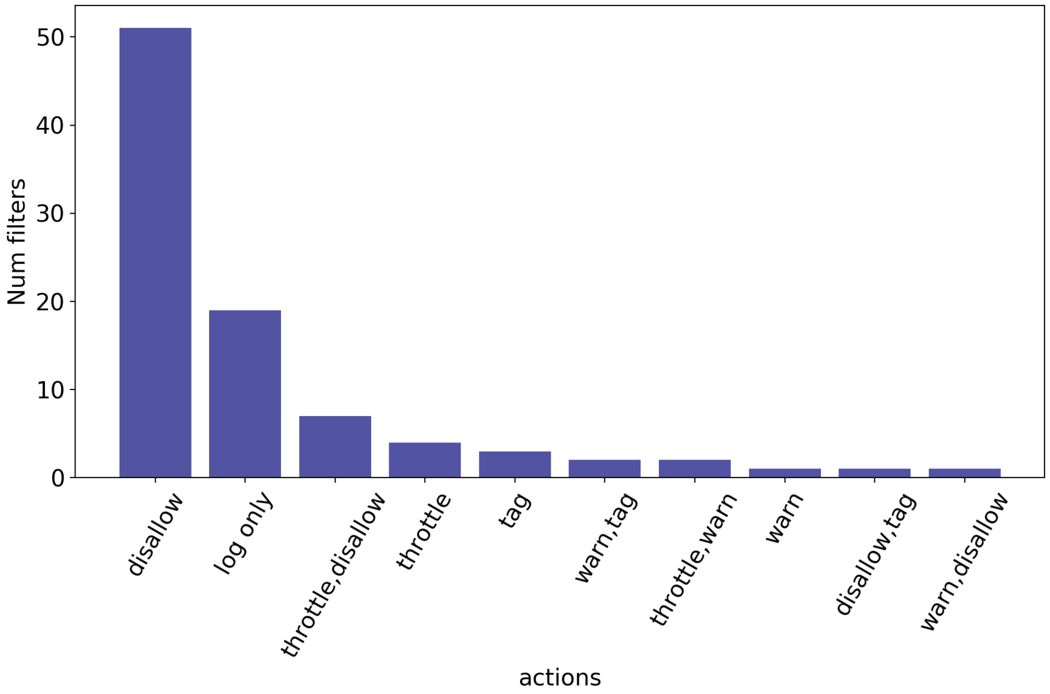


Figure 5.4.: EN Wikipedia edit filters: Filters actions for enabled hidden filters (January 2019)

distribution of manually assigned codes and their parent categories can be view on figure 5.7.

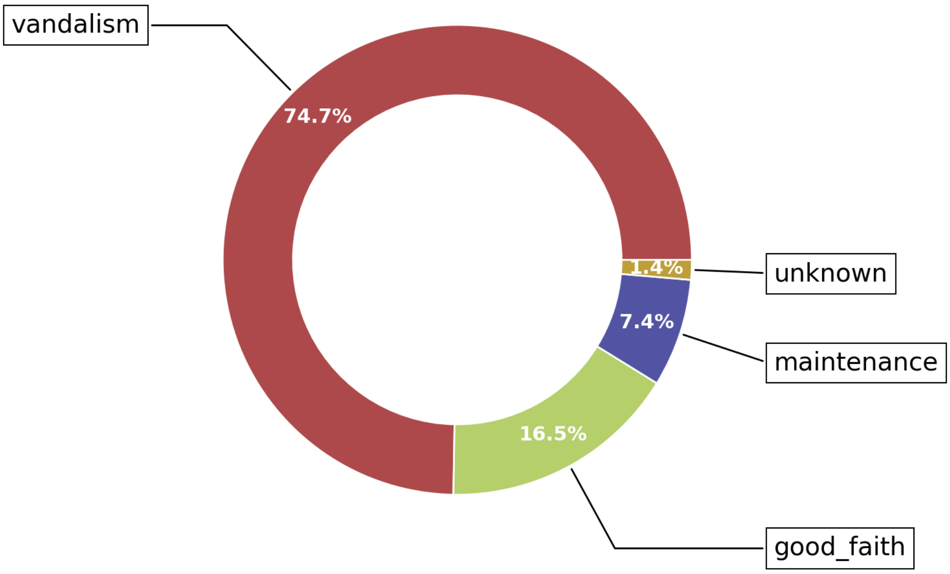


Figure 5.5.: Manual tags parent categories distribution: all filters

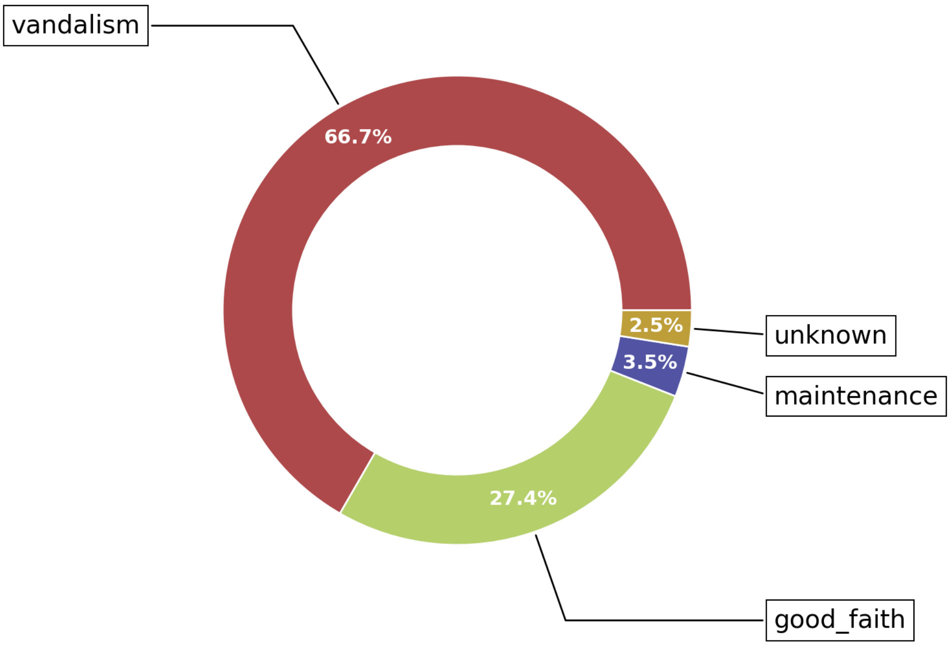


Figure 5.6.: Manual tags parent categories distribution: enabled filters (January 2019)

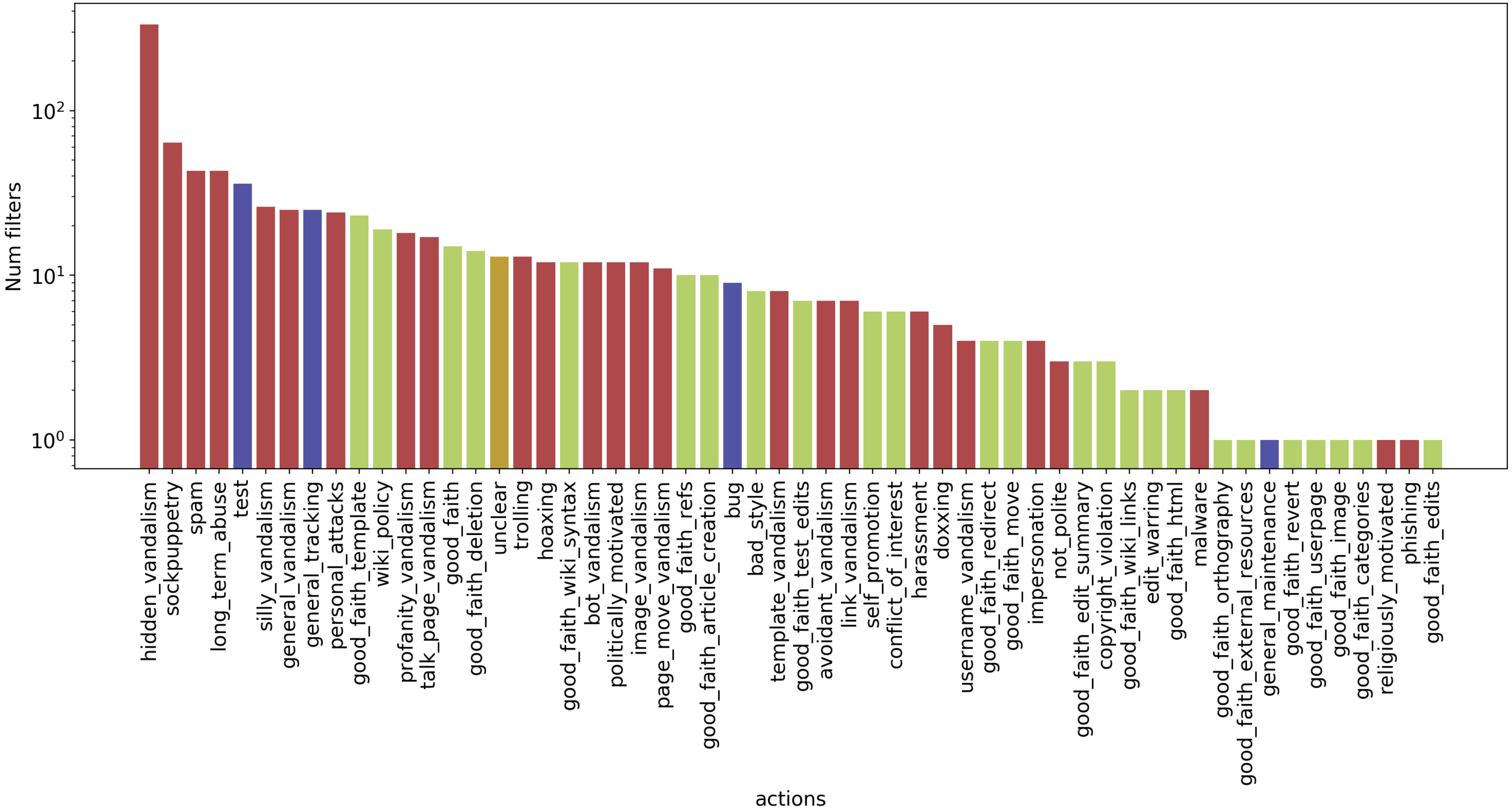


Figure 5.7.: Edit filters manual tags distribution

Another feature explored was the explicit targeting of not confirmed users (see table 5.1). It arrests attention that various filters have what the edit filter managers have dubbed "the newbie check": `!("confirmed" in user_groups)` as one of their first conditions. There are in total 43 such filters, 26 of them are enabled as of January 2019 (so they make up approximately 20% of all enabled filters at the time) and 9 of the enabled filters disallow the edit directly when matched.

Filter ID	Publicly available description	Hitcount	Actions
61	New user removing references	1611956	tag
384	Addition of bad words or other vandalism	1159239	disallow
30	Large deletion from article by new editors	840871	warn,tag
636	Unexplained removal of sourced content	726764	warn
3	New user blanking articles	700522	warn,tag
432	Starting new line with lowercase letters	558578	warn,tag
225	Vandalism in all caps	482872	disallow
50	Shouting	480960	warn,tag
231	Long string of characters containing no spaces	380302	warn,tag
46	”Poop” vandalism	356945	disallow
39	School libel and vandalism	150568	warn,tag
11	You/He/She/It sucks	109657	warn,tag
680	Adding emoji unicode characters	95242	disallow
365	Unusual changes to featured or good content	85470	disallow
126	Youtube links	65137	log only
803	Prevent new users from editing other's user pages	46756	disallow
117	removal of Category:Living people	43822	tag
113	Misplaced #redirect in articles	20885	warn,tag
59	New user removing templates on image description	19938	tag
655	Large plot section addition	16051	tag
784	Harambe vandalism	9265	disallow
912	Possible ”fortnite” vandalism	7505	warn,tag
860	Ryan Ross vandalism	3451	disallow
766	Alt-right labeling	1866	warn,tag
921	Suspicious claims of nazism	1422	tag
843	Prevent new users from creating redirects to [[Donald Trump]]	98	disallow

Table 5.1.: Filters aimed at unconfirmed users

5.3.5. Who Trips Filters

As of 15 March 2019 $16,489,266$ of the filter hits were caused by IP users, whereas logged in users had matched an edit filter's pattern $6,984,897$ times.

A lot of the logged in users have newly created accounts (many filters look for newly created, or respectively, not confirmed accounts in their pattern).

A user who just registered an account (or who doesn't even bother to) is rather to be expected to be inexperienced with Wikipedia, not familiar with all policies and guidelines and perhaps nor with MediaWiki syntax.

It also sounds plausible that majority of vandalism edits come from the same type of newly/recently registered accounts. In general, it is rather unlikely that an established Wikipedia editor should at once jeopardise the encyclopedia's purpose and start vandalising. Although apparently there are determined trolls who "work accounts up" to admin and then run rampant.

5.4. Filter Activity

This section explores filter activity from two perspectives: It looks into the numbers of filter hits per month in 5.4.1 and discusses the most active filters over the years in 5.4.2.

5.4.1. Filter Hits per Month

The number of filter hits per month over the years can be backtracked on figure 5.8. There is a dip in the number of hits in late 2014 and quite a surge in the beginnings of 2016, after which the overall number of filter hits stayed higher. There is also a certain periodicity to the graph, with smaller dips in the northern hemisphere's summer months (June, July, August) and smaller peaks in autumn/winter (mostly October/November). This tendency is not observed for the overall number of edits (see figure 4.6). Apparently, above all editors tripping filters are on vacation in June, July and August.

Further, it is interesting to break down filter activity according to the types determined via the manual tagging (see section 5.2): The corresponding distribution is shown in figure 5.9. On the one hand, it demonstrates above all a surge in the hits of filters targeting vandalism in 2016. On the other hand, another, somewhat subtler trend emerges: In the first years following the introduction of the mechanism, good faith filters were matched most frequently. This changed around the end of 2012 and since then the most hits are marked by vandalism filters.

Regarding the hits surge and subsequent higher hit numbers, three possible explanations come to mind:

1. the filter hits mirror the overall edits pattern from this time;
2. there was a general rise in vandalism in this period;
3. or there was a change in the edit filter software that allowed more filters to be activated, or a bug that caused the peak (in the form of a lot of false positives).

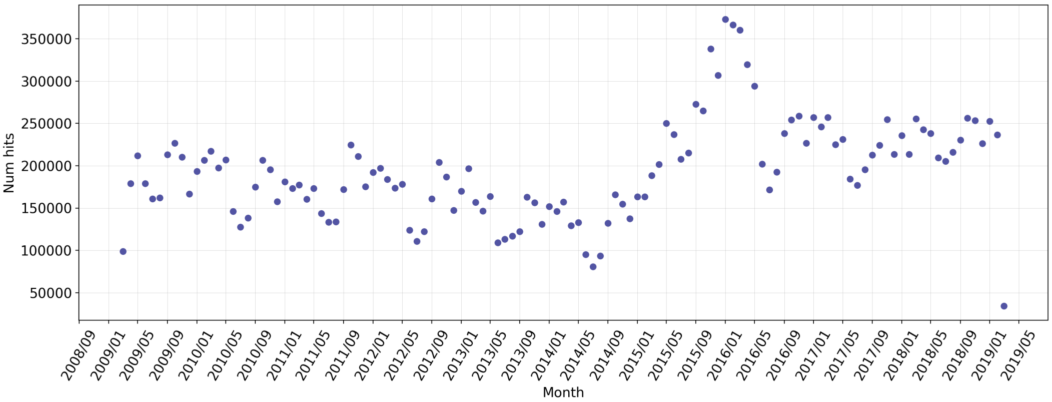


Figure 5.8.: EN Wikipedia edit filters: Hits per month

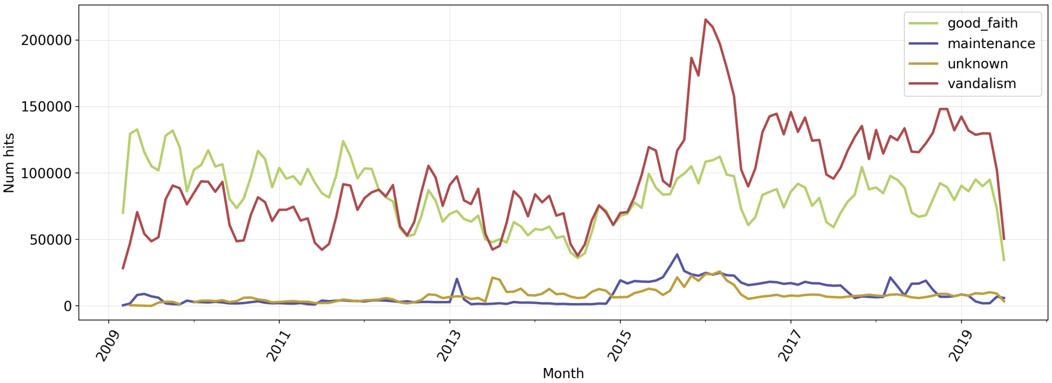


Figure 5.9.: EN Wikipedia edit filters: Hits per month according to manual tags

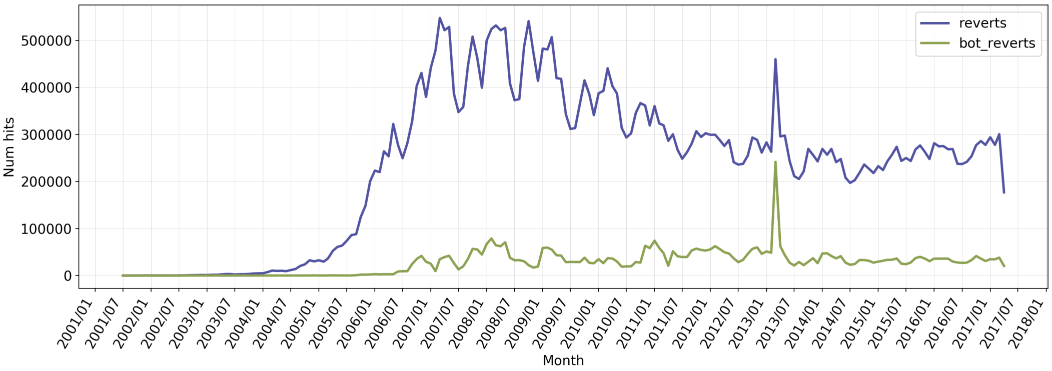


Figure 5.10.: EN Wikipedia: Reverts for July 2001–April 2017

I've undertaken following steps in an attempt to verify or refute each of these speculations:

The filter hits mirror the overall edits pattern from this time
I've compared the filter hits pattern with the overall number of edits of the time (May 2015–May 2016). No correspondence could be determined (see figure 4.6).

There was a general rise in vandalism in this period
This assumption is supported by the peak in the hits of vandalism related filters end 2015–beginning 2016 observed in figure 5.9. In order to verify it, a comparison of the filters' hits patterns with revert patterns of other quality control mechanisms seems logical. Unfortunately, computing these numbers is time-consuming and not completely trivial. One needs a dump of English Wikipedia's edit history data for the period in question; then one has to determine the reverts in this data set (e.g. by using the *mwreverts* python library); and then, more specifically, one needs to extract reverts done by quality control actors. Last step is crucial, since not every revert signifies a malicious edit is being reverted. This point is aptly illustrated by [GH17] who have demonstrated that reverts can mean productive collaborative work between different agents.

The dumps are large and it takes time and computing power to obtain them and extract reverts. According to Geiger and Halfaker who have done this for their replication study [GH17], the April 2017 database dump offered by the Wikimedia Foundation was 93GB compressed and it took a week to extract reverts out of it on a 16 core Xeon workstation. They also list the challenges they faced in determining bot accounts and their reverts.

Since time was scarce, I have run a first check of this assumption using the 2017 reverts dataset compiled by Geiger and Halfaker's for their study [4]. The dataset is old, but still sufficient for scrutinising events at the beginning of 2016. Figure 5.10 shows the total number of reverts, as well as reverts done by bots over time computed by Geiger and Halfaker. The filter hits pattern of 2015–2016 with the peak in filter hits and subsequent higher number of overall hits is not mirrored by the revert numbers [5] (note that the y-axis of both the revert and the filter hit plots is of the same magnitude). As cautioned earlier, not every revert can be equated with cleaning up a disruptive edit, however, figure 5.10 demonstrates that either quality control reverts constitute a relatively small portion of all reverts being done, or that there wasn't a general surge in vandalism around this time. (Or that only vandalism caught by filters

[4] Both researchers have placed a great value on reproducibility and have published their complete datasets, as well as scripts they used for their analyses for others to use and verify: https://github.com/halfak/are-the-bots-really-fighting.

[5] Just for completenes, the spike in March 2013 is the batch action by AddBot removing interwiki links, since these were handled by Wikidata discussed in the introduction of Geiger and Halfaker's paper. It didn't have anything to do with vandalism.

peaked, which sounds somewhat improbable.)

There was a change in the edit filter software that allowed more filters to be activated, or a bug that caused false positives

Since so far neither of the other hypothesis could be verified, this explanation sounds likely. Another piece of data that seems to support it is the breakdown of the filter hits according to triggered filter action. As demonstrated on figure 5.11, there was above all a significant hits peak caused by “log only” filters. As discussed in section 4.5.1, it is an established praxis to introduce new filters in “log only” mode and only switch on additional filter actions after a monitoring period showed that the filters function as intended. Hence, it is plausible that new filters in logging mode were introduced, which were then switched off after a significant number of false positives occurred. However, upon closer scrutiny, this could not be confirmed. The filters with greatest number of hits in the period January–March 2016 are mainly the most triggered filters of all times and nearly all of them have been around for a while in 2016. Also, no bug or a comparable incident with the software was found upon an inspection of the extension’s issue tracker [Pla16a], or commit messages of the commits to the software done during May 2015–May 2016 [Gar19a]. Moreover, no mention of the hits surge was found in the noticeboard [Wik19y] and edit filter talk page archives [Wik19ae]. The in section 5.4 mentioned condition limit has not changed either, as far as I can tell from the issue tracker, the commits and discussion archives, so the possible explanation that simply more filters have been at work since 2016 seems to be refuted as well.

The only somewhat interesting pattern that seems to shed some light on the matter is the breakdown of hits according to the editor’s action which triggered them: There is an obvious surge in the attempted account creations in the period November 2015–May 2016 (see figure 5.12). As a matter of fact, this could also be the explanation for the peak of log only hits—the most frequently tripped filter for the period January–March 2016 is filter 527 “T34234: log/throttle possible sleeper account creations”. It is a throttle filter, with no further actions enabled, so every time an edit matches its pattern, a “log only” entry is created in the abuse log. And the 3rd most active filter is a “log only” filter as well: 650 “Creation of a new article without any categories”. (It was neither introduced at the time, nor was there any major change in the filter pattern.) Together, filters 527 and 650 are responsible for over 60% of the “log only” hits in every of the months January, February and March 2016.

Another idea that seemed worth pursuing was to look into the editors who tripped filters and their corresponding edits. For the period January–March 2016 there are some very active IP editors, the top of whom (with over 1.000 hits) seemed to be engaging exclusively in the (probably automated) posting of spam links. Their edits however constitute some 1-3% of all hits from the period which is insufficient to explain the peak [6]. A more systematic

[6] Upon closer examination, these edits all seemed to contain spam links about erectile

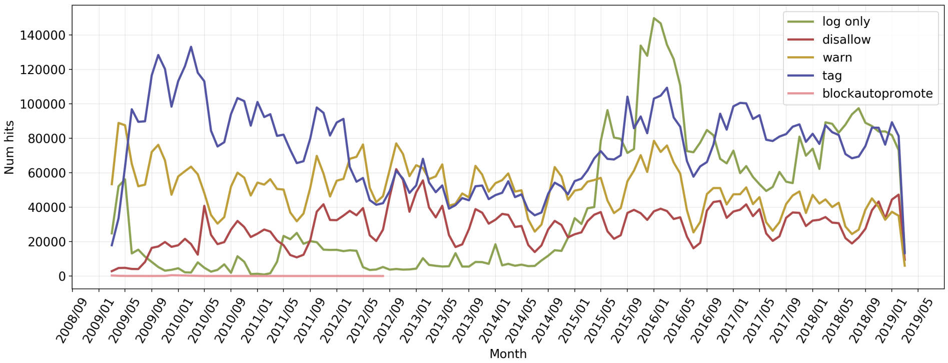


Figure 5.11.: EN Wikipedia edit filters: Hits per month according to filter action

scrutiny of the editors causing the hits was not possible due to time constraints, but may contribute more insights. Right now, all the data analysed on the matter stems from the *abuse_filter_log* table and the checks of the content of the edits were done manually on a sample basis via the web frontend of the AbuseLog [Wik19e] where one can click on the diff of the edit for edits that matched public filters. No simple automated check of what the offending editors were trying to publish was possible since the *abuse_filter_log* table does not store the text of the edit which matches a filter's pattern directly, but rather contains a reference to the *text* table where the wikitext of all individual page revisions is stored [Wik19bc]. One needs to join the hit data from *abuse_filter_log* with the *text* table to obtain the content of the edits.

Last but not least, an investigation into the pages on which the filters were triggered proved them (the pages) to be quite innocuous: The page where most filter hits were logged in January 2016 (beside the login page, on which all account creations are logged) was "Skateboard" and the 660 filter hits here are rather insignificant compared to the 372.907 hits for the whole month. And the page in March (apart from the user login page) on which most filter hits took place was the user page for user 209.236.119.231 who was also the editor with second most hits and who was apparently trying to post spam links on his own user page (after posting twice to "Skateboard"). In general, the pages on which filters match seem more like a randomly selected platform on which the disrupting editors unload their spam.

5.4.2. Most Active Filters Over the Years

Table 5.2 displays the ten most active filters of all times together with their corresponding number of hits, actions, and manually assigned label. Only one

dysfunction medication and their IP records pertained to a Russian registry. It is however possible that the offending editors were using a VPN or another proxy technology. The speculations about the intent of the edits remain out of the scope of the present work.

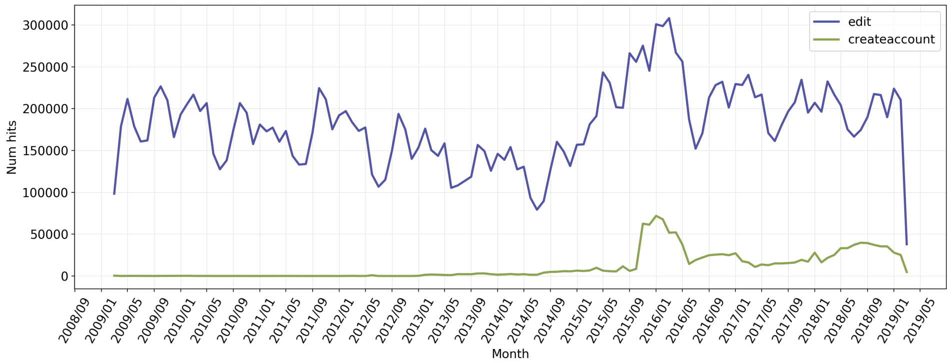


Figure 5.12.: EN Wikipedia edit filters: Hits per month according to triggering editor's action

among them fits the description of targeting malicious determined vandals: filter 527 "T34234: log/throttle possible sleeper account creations". The second area in which these filters are active are various types of blankings (mostly by new users) where the filters issue warnings pointing towards possible alternatives the editor may want to achieve or the proper procedure for deleting articles for instance. The table also shows that the mechanism ended up being quite active in preventing silly (e.g. inserting series of repeating characters) or profanity vandalism.

It is also interesting to trace the trends for the ten most active filters for each year since the introduction of the AbuseFilter extension. According to tables 5.3 through 5.12, this list has remained remarkably stable over time: From year to year, there is a difference of 2-3 filters. Also, at least half of the most active filters for each year overlap with the most active filters of all times.

5.5. Conclusions

This chapter explored the edit filters on EN Wikipedia in order to determine what types of tasks these filters take over, and how these tasks have evolved over time.

Different characteristics of the edit filters, as well as their activity through the years were scrutinised. Three main types of filter tasks were identified: preventing/tracking vandalism, guiding good faith but nonetheless disruptive edits towards a more constructive contribution, and various maintenance jobs such as tracking bugs or other conspicuous behaviour. It was further observed, that filters aimed at particularly malicious users or behaviours are usually hidden, whereas filters targeting general patterns are viewable by anyone interested. It was determined that hidden filters seem to fluctuate more, which makes sense given their main area of application. Public filters often target silly vandalism or test type edits, as well as spam. Disallowing edits by very determined vandals handled by hidden filters are in accord with the initial

Filter ID	Hitcount	Publicly available description	Actions	Manual tag (parent category)
61	1,611,956	new user removing references	tag	good_faith_refs (good_faith)
135	1,371,361	repeating characters	tag, warn	silly_vandalism (vandalism)
527	1,241,576	T34234: log/throttle possible sleeper account creations (hidden filter)	throttle	sockpuppetry (vandalism)
384	1,159,239	addition of bad words or other vandalism	disallow	profanity_vandalism (vandalism)
172	935,925	section blanking	tag	good_faith_deletion (good_faith)
30	840,871	large deletion from article by new editors	tag, warn	good_faith_deletion (good_faith)
633	808,716	possible canned edit summary	tag	general_vandalism (vandalism)
636	726,764	unexplained removal of sourced content	warn	good_faith_deletion (good_faith)
3	700,522	new user blanking articles	tag, warn	good_faith_deletion (good_faith)
650	695,601	creation of a new article without any categories	(log only)	general_tracking (maintenance)

Table 5.2.: What do most active filters do?

aim with which the filters were introduced (compare section 4.4). The high number of such filters (compare section 5.3.4) seems to confirm that edit filters are fulfilling their purpose. On the other hand, when the ten most active filters of all times (see table 5.2) are regarded, only one of them appears to take care of the malicious determined vandals who motivated the creation of the AbuseFilter extension. The rest of the most frequently matching filters target a combination of good faith edits (above all such concerning deletions) and silly/profanity vandalism. Interestingly, that is not what the developers of the extension believed it was going to be good for: “It is not, as some seem to believe, intended to block profanity in articles (that would be extraordinarily dim), nor even to revert page-blankings, ” claimed its core developer on 9 July 2008 [Wik19ad].

A further assumption that didn't carry into effect was that “filters in this extension would be triggered [7] fewer times than once every few hours” [Wik19ac].

[7] Here, by “trigger” is meant that an editor's action will match a filter's pattern and set off the configured filter's action(s).

Filter ID	Publicly available description	Hitcount
135	repeating characters	175455
30	”large deletion from article by new editors”	160302
61	”new user removing references”	147377
18	Test type edits from clicking on edit bar	133640
3	”new user blanking articles”	95916
172	”section blanking”	89710
50	”shouting” (contribution consists of all caps, numbers and punctuation)	88827
98	”creating very short new article”	80434
65	”excessive whitespace”	74098
132	”removal of all categories”	68607

Table 5.3.: 10 most active filters in 2009

Filter ID	Publicly available description	Hitcount
61	”new user removing references”	245179
135	repeating characters	242018
172	”section blanking”	148053
30	”large deletion from article by new editors”	119226
225	Vandalism in all caps	109912
3	”new user blanking articles”	105376
50	”shouting”	101542
132	”removal of all categories”	78633
189	BLP vandalism or libel	74528
98	”creating very short new article”	54805

Table 5.4.: 10 most active filters in 2010

Filter ID	Publicly available description	Hitcount
61	”new user removing references”	218493
135	repeating characters	185304
172	”section blanking”	119532
402	New article without references	109347
30	Large deletion from article by new editors	89151
3	”new user blanking articles”	75761
384	Addition of bad words or other vandalism	71911
225	Vandalism in all caps	68318
50	”shouting”	67425
432	Starting new line with lowercase letters	66480

Table 5.5.: 10 most active filters in 2011

Filter ID	Publicly available description	Hitcount
135	repeating characters	173830
384	Addition of bad words or other vandalism	144202
432	Starting new line with lowercase letters	126156
172	”section blanking”	105082
30	Large deletion from article by new editors	93718
3	”new user blanking articles”	90724
380	Multiple obscenities	67814
351	Text added after categories and interwiki	59226
279	Repeated attempts to vandalize	58853
225	Vandalism in all caps	58352

Table 5.6.: 10 most active filters in 2012

Filter ID	Publicly available description	Hitcount
135	repeating characters	133309
384	Addition of bad words or other vandalism	129807
432	Starting new line with lowercase letters	94017
172	”section blanking”	92871
30	Large deletion from article by new editors	85722
279	Repeated attempts to vandalize	76738
3	”new user blanking articles”	70067
380	Multiple obscenities	58668
491	Edits ending with emoticons or !	55454
225	Vandalism in all caps	48390

Table 5.7.: 10 most active filters in 2013

Filter ID	Publicly available description	Hitcount
384	Addition of bad words or other vandalism	111570
135	repeating characters	111173
279	Repeated attempts to vandalize	97204
172	”section blanking”	82042
432	Starting new line with lowercase letters	75839
30	Large deletion from article by new editors	62495
3	”new user blanking articles”	60656
636	Unexplained removal of sourced content	52639
231	Long string of characters containing no spaces	39693
380	Multiple obscenities	39624

Table 5.8.: 10 most active filters in 2014

Filter ID	Publicly available description	Hitcount
650	Creation of a new article without any categories	226460
61	New user removing references	196986
636	Unexplained removal of sourced content	191320
527	T34234: log/throttle possible sleeper account creations	189911
633	Possible canned edit summary	162319
384	Addition of bad words or other vandalism	141534
279	Repeated attempts to vandalize	110137
135	repeating characters	99057
686	IP adding possibly unreferenced material to BLP	95356
172	”section blanking”	82874

Table 5.9.: 10 most active filters in 2015

Filter ID	Publicly available description	Hitcount
527	T34234: log/throttle possible sleeper account creations	437099
61	New user removing references	274945
650	Creation of a new article without any categories	229083
633	Possible canned edit summary	218696
636	Unexplained removal of sourced content	179948
384	Addition of bad words or other vandalism	179871
279	Repeated attempts to vandalize	106699
135	repeating characters	95131
172	”section blanking”	79843
30	Large deletion from article by new editors	68968

Table 5.10.: 10 most active filters in 2016

Filter ID	Publicly available description	Hitcount
61	New user removing references	250394
633	Possible canned edit summary	218146
384	Addition of bad words or other vandalism	200748
527	T34234: log/throttle possible sleeper account creations	192441
636	Unexplained removal of sourced content	156409
650	Creation of a new article without any categories	151604
135	repeating characters	80056
172	”section blanking”	70837
712	Possibly changing date of birth in infobox	59537
833	Newer user possibly adding unreferenced or improperly referenced material	58133

Table 5.11.: 10 most active filters in 2017

Filter ID	Publicly available description	Hitcount
527	T34234: log/throttle possible sleeper account creations	358210
61	New user removing references	234867
633	Possible canned edit summary	201400
384	Addition of bad words or other vandalism	177543
833	Newer user possibly adding unreferenced or improperly referenced material	161030
636	Unexplained removal of sourced content	144674
650	Creation of a new article without any categories	79381
135	repeating characters	75348
686	IP adding possibly unreferenced material to BLP	70550
172	”section blanking”	64266

Table 5.12.: 10 most active filters in 2018

As a matter of fact, a quick glance at the AbuseLog [Wik19e] confirms that there are often multiple filter hits per minute, so the mechanism is used fairly actively, despite that its areas of application partially diverge from the ones initially conceived. In fact, the numbers of filter hits on EN Wikipedia are in the same order of magnitude as the revert numbers (compare figures 5.8 and 5.10).

Regarding the temporal filter activity trends, it was ascertained that a sudden peak took place in the end of 2015–beginnings of 2016, after which the overall filter hit numbers stayed higher than they used to be before this occurrence. Although there were some pointers towards what happened there: a surge in account creation attempts and possibly a big spam wave (the latter has to be verified in a systematic fashion), no really satisfying explanation of the phenomenon could be established. This remains one of the possible direction for future studies.

In their 2012 paper Halfaker and Riedl propose a bot taxonomy according to which Wikipedia bots could be classified in one of the following task areas: content injection, monitoring, or curating; augmenting MediaWiki functionality; or protection from malicious activity [HR12]. And although there are no filters that inject or curate content, there are definitely filters whose aim is to protect the encyclopedia from malicious activity, and such that augment MediaWiki's functionality e.g. by providing warning messages (with hopefully helpful feedback) or by tagging certain behaviours to be aggregated on dashboards for later examination. In this sense, edit filters and bots appear to be rather similar.

6. Discussion

I started this inquiry with following questions:
Q1: What is the role of edit filters among existing algorithmic quality-control mechanisms on Wikipedia (bots, semi-automated tools, ORES, humans)?
Q2: Edit filters are a classical rule-based system. Why are they still active today when more sophisticated ML approaches exist?
Q3: Which type of tasks do filters take over?
Q4: How have these tasks evolved over time (are there changes in the type, number, etc.)?

In what follows, I go over each of them and summarise the findings.

6.1. Q1 What is the role of edit filters among existing quality-control mechanisms on Wikipedia (bots, semi-automated tools, ORES, humans)?

When edit filters were introduced in 2009, various other mechanisms that took care of quality control on Wikipedia had already been in place for some time. However, the community felt the need for an instrument for preventing easy to recognise but pervasive and difficult to clean up vandalism as early as possible. This was supposed to take workload off the other mechanisms along the quality control process (see figure 4.7), especially off human editors who could then use their time more productively elsewhere, namely to check less obvious cases.

Both filters and bots are completely automated mechanisms, thus a comparison between the two seems reasonable. What did the filters accomplish differently? A key distinction is that while bots check already published edits which they may decide to eventually revert, filters are triggered before an edit ever published. One may argue that nowadays this is not a significant difference. Whether a disruptive edit is outright disallowed or caught and reverted two seconds after its publication by ClueBot NG doesn't have a tremendous impact on the readers: The vast majority of them will never see the edit either way. Still, there are various examples of vandalism that didn't survive long on Wikipedia but the brief time before they were reverted was sufficient for hundreds of media outlets to report these as news [Eld16], which severely undermines the project's credibility.

Another difference between bots and filters underlined several times in community discussions was that as a MediaWiki extension edit filters are part of the core software whereas bots are running on external infrastructure which makes them both slower and generally less reliable. (Compare Geiger's account about running a bot on a workstation in his apartment which he simply pulled the plug on when he was moving out [Gei14].) Nowadays, we can ask ourselves whether this is still of significance: A lot of bots are run on Toolforge [Wik19c], a cloud service providing a hosting environment for a variety of applications (bots, analytics, etc.) run by volunteers who work on Wikimedia projects. The service is maintained by the Wikimedia Foundation the same way the Wikipedia servers are, so it is in consequence just as reliable and available as the encyclopedia itself. The argument that someone powered off the basement computer on which they were running bot X is just not as relevant anymore.

When comparing the tasks of bots proposed in related work (chapter 2) with the content analysis of filters' tasks conducted in chapter 5 (see also discussion for Q3 in section 6.3), the results show great overlaps between the tasks descriptions for both tools. From an end result perspective it doesn't seem to make a big difference, whether a problem is taken care of by an edit filter or a bot. As mentioned in the paragraph above, whether malicious content is directly disallowed or reverted two seconds later (in which time probably a total of three users have seen it if any) is hardly a qualitative difference for Wikipedia's readers. I would argue though that there are other stakeholders for whom the choice of mechanism makes a bigger difference: the operators of the quality control mechanisms and the users whose edits are being targeted. The significant distinction for operators is that the architecture of the edit filter plugin supposedly fosters collaboration which results in a better system (compare with the famous "given enough eyeballs, all bugs are shallow" [Ray99]). Any edit filter manager can modify a filter causing problems and the development of a single filter is usually a collaborative process. Just a view on the history of most filters reveals that they have been updated multiple times by various users. In contrast, bots' source code is often not publicly available and they are mostly run by one operator only, so no real peer review of the code is practiced and the community has time and again complained of unresponsive bot operators in emergency cases [Wik19ac]. (On the other hand, more and more bots are based on code from various bot development frameworks such as pywikibot [pyw], so this is not completely valid either.) At the same time, it seems far more difficult to become an edit filter manager: There are only very few of them, the vast majority admins or in exceptional cases very trusted users. By contrast, a bot operator only needs an approval for their bot by the Bot Approvals Group and can get going.

The choice of mechanism also makes a difference for the editor whose edits have been deemed disruptive. Filters assuming good faith seek communication with the offending user by issuing warnings which provide some feedback and allow the user to modify their edit (hopefully in a constructive fashion) and publish it again. Bots on the other hand revert everything their algorithms find malicious directly. They also leave warning messages on the user's talk page informing them that their edits have been reverted because the bot's heuristic was matched and point them to a false positives page where they can make a report. It is still a revert-first-ask-questions-later approach which is rather discouraging for good faith newcomers. In case of good faith edits, this would mean that an editor wishing to dispute this decision should raise the issue on the bot's talk page and research has shown that attempts to initiate discussions with (semi-)automated quality control agents have in general quite poor response rates [HGMR13].

Compared to MediaWiki's page protection mechanism, edit filters allow for accurate control on user level: One can implement a filter targeting specific malicious users directly instead of restricting edit access for everyone.

6.2. Q2: Edit filters are a classical rule-based system. Why are they still active today when more sophisticated ML approaches exist?

Research has long demonstrated higher precision and recall of machine learning methods [PSG08]. With this premise in mind, one has to ask: Why are rule based mechanisms such as the edit filters still widely in use? Several explanations of this phenomenon sound plausible. For one, Wikipedia's edit filters are an established system which works and does its work reasonably well, so there is no need to change it ("never touch a running system"). Secondly, it has been organically woven in Wikipedia's quality control ecosystem. There were historical necessities to which it responded and people at the time believed the mechanism to be the right solution to the problem they had. We could ask why was it introduced in the first place when there were already other mechanisms. Beside the specific instances of disruptive behaviour stated by the community as motivation to implement the extension, a very plausible explanation here is that since Wikipedia is a volunteer project a lot of stuff probably happens because at some precise moment there are particular people who are familiar with some concrete technologies so they construct a solution using the technologies they are good at using (or want to use).

Another interesting reflection is that rule based systems are arguably easier to implement and above all to understand by humans which is why they still enjoy popularity today. On the one hand, overall less technical knowledge is required in order to implement a single filter: An edit filter manager has to "merely" understand regular expressions. Bot development by contrast is a little more challenging: A developer needs reasonable knowledge of at least one programming language and on top of that has to make themself familiar with artefacts like the Wikimedia API. Moreover, since regular expressions are still somewhat human readable and comprehensible unlike a lot of popular machine learning algorithms, it is easier to hold rule based systems and their developers accountable. Filters are a simple mechanism (simple to implement) that swiftly takes care of cases that are easily recognisable as undesirable. ML needs training data (which expensive), and it is not simple to implement. What is more, rule based mechanisms allow for a finer granularity of control:

An edit filter can define a rule to explicitly exclude particular malicious users from publishing, which cannot be straightforwardly implemented in a machine learning algorithm.

6.3. Q3: Which type of tasks do filters take over?

Chapter 5 shows that edit filters target juvenile and grave vandalism, spam, good faith disruptive edits (e.g. blanking an article instead of moving it because of unfamiliarity with the software and proper procedure), and maintenance tasks. In total, 2/3 of the filters ever implemented are still hidden, and since according to the guidelines filters are supposed to be hidden when aimed at egregious vandalism by specific malicious users [Wik19r], the AbuseFilter extension appears to be used in accordance with its declared purpose. At the same time, the January 2019 snapshot of the *abuse_filter* database table revealed a nearly equal numbers of enabled public and private filters. This means that at the time, filters were targeting specific vandals as much as general disruptive behaviour. It also leads to the conclusion that hidden filters fluctuate more which seems reasonable given their application area: specific users and behaviours.

As demonstrated by the bot taxonomy proposed by Halfaker and Riedl [HR12] referred to in section 5.5, bots are also doing a lot of these or similar tasks. So, when a new problem arises, how does the community decide whether to implement a bot or a filter to handle it? As discussed in the previous section 6.3, this probably partially depends on who discovers/takes care of the problem and what technology they are familiar with and have access to. There are also some guidelines (compare section 4.5.1) which underline that filters are most suitable for problems concerning all pages and point out different approaches for solving other issues: using page protection for problems with a single page; using the title and spam blacklist for persistent spam waves or attempts to create abusive titles; using bots for in depth checks or problems with a single page. Moreover, it is stated that no trivial formatting mistakes should trip filters[Wik19ay]: This seems like a waste of computing power and unnecessary irritation to the user. For what it is worth, I also think that bots are more suitable to take care of such cases. However, the community is not always consistently sticking to these guidelines, and they do occasionally implement filters that contradict them. (Examples therefor are filters that target non-disruptive or non-problematic behaviour such as filter 308 “Malformed Mediation Cabal Requests”, or the fairly frequent hiding of filters tracking general behaviour.) These are mostly switched off or in the case of hiding general patterns—made public—again relatively fast but there are also examples such as the filter 432 “Starting new line with lowercase letters” (still active as of 24 July 2019) which in my opinion violates the above mentioned trivial mistakes rule.

As a matter of fact, multiple edit filter managers also run bots. Therefore, it looks relevant to consider how they decide which mechanism to apply when faced with a particular issue. Preliminary results have shown that some of the users concerned seem to be rather bot operators who implement auxiliary filters and some—primarily edit filter managers who implement auxiliary bots. As mentioned in section 6.6, future work could further explore the relationships of filters and bots implemented by the same user, especially by taking the (currently unavailable) *abuse_filter_history* table into account, or by conducting interviews with the users in question.

At the end, closer scrutiny and critical evaluation of the filter patterns are required. It can be discussed whether it is fair and justified that 20% of the enabled filters target only new (not confirmed) editors. Why is it all right for an established editor to use swear words (see filter 384 "Addition of bad words or other vandalism") or insert longer strings of all caps (filter 50 "Shouting") whereas it is not for newbies?

6.4. Q4: How have these tasks evolved over time (are there changes in the type, number, etc.)?

Following insights about temporal trends were uncovered: Firstly, edit filters have been much more active than the initially anticipated few hits per hour—a consultation of the AbuseLog shows several entries per minute and the hit numbers resemble the revert counts in order of magnitude. Secondly, the list of most active filters of all times reveals above all older filters which continue to be matched very frequently. Moreover, it is mostly the same old filters which have been highly active through the years: The list of the ten most active filters for each year since the introduction of the AbuseFilter extension is fairly stable. Although, as pointed out in section 4.2, filter patterns can be changed, they are mostly only optimised for efficiency, so it can be assumed filters have been catching the same troublesome behaviour over the years. Thirdly, the overall number of filter hits has risen since 2016 when a somewhat puzzling spike in filter hits which needs future investigation took place. Additionally, the general tendency is that over time less good faith filter hits and more vandalism related ones occurred.

All in all, beside the peak in hits from 2016, additional temporal patterns of filters characteristics and activity can be explored. These include practices of filters' creation, configuration, and refactoring.

6.5. Limitations

This work presents an initial attempt at analysing Wikipedia's edit filter system; as such, it has several limitations.

6.5.1. Limitations of the Data

Firstly, the thesis focuses on English Wikipedia only. This offers an excellent starting point for the analysis of edit filters: After all, EN Wikipedia was the first language version to which the mechanism was introduced. However, valuable lessons can be learnt—about the communities, models of governance, usefulness of filters, etc.—from comparing edit filters' usage and activity across different language versions. Just recall how for instance the edit filter managers group doesn't exist in certain language versions (section 4.5.2) and instead there it is administrators who have an *abusefilter-modify* permission next to their other rights. Effectively, for these language versions of Wikipedia (the Spanish, German, and Russian ones), a much bigger group of users has access to the mechanism. It is expected that this shapes its governance and usage patterns.

Moreover, the *abuse_filter_history* table was not available, so no systematic analysis of the filters' development over time could be realised (see section 5.1).

Finally, I had no access to the details of hidden filters, so no investigation of their patterns (for instance verifying whether they really target specific users) was possible.

6.5.2. Limitations in the Research Process

Unfortunatly, conducting a classic ethnographic analysis was not possible. It would have been particularly insightful to talk to edit filter managers (above all such who are simultaneously also bot operators) and developers of the extension, as well as regular editors who have tripped a filter about their experiences. Basically, really only "found data" was used, and as pointed out in section 3.1 this has the shortcoming of observing only what was discussed in the documentation archives and recorded by the logs. As Geiger and Halfaker maintain, Wikipedia's databases have the purpose of allowing the Wikipedian community to build an encyclopedia, not to facilitate scientific research [GH17]. Future studies can and should use further data sources and for instance utilise the first insights of the current research as interview prompts.

Another limitation that comes to mind is related to the applied methodology of trace ethnography. The data of the present study do not speak for themselves: Instead, domain knowledge of the Wikipedian ecosystem is necessary in order to be able to accurately invert traces. Previous to this research, I have had a Wikipedia account for several years but have only used it to make occasional (rather minor) edits. I have learnt a lot since the beginning of the project, but it is still very much possible that I have misinterpreted data due to insufficient experience and lack of background knowledge.

Thirdly, as signaled in section 5.2, the manual filter classification was undertaken by one person only (me), so my biases have certainly shaped the labels. To increase reliability, the coding process should be applied by at least one more researcher and both sets of labeled data should be compared.

6.6. Directions for Future Studies

Throughout the thesis, a variety of intriguing questions arose which couldn't be addressed due to various reasons, above all—insufficient time. Here, a comprehensive list of all these pointers for possible future research is provided.

1. **How have edit filters's tasks evolved over time?** Unfortunately, no detailed historical analysis of the filters could be realised, since the database table storing changes to individual filters (*abuse_filter_history*) is not currently replicated (see section 5.1). As mentioned in section 5.1, a patch aiming to renew the replication of the table is currently under review [Mes19]. When a dump becomes available, an extensive investigation of filters' actions, creation and activation patterns, as well as patterns they have targeted over time will be possible.

2. **What proportion of quality control work do filters take over?** Filter hits can be systematically compared with the number of all edits and reverts via other quality control mechanisms.

3. **Is it possible to study the filter patterns in a more systematic fashion? What can be learnt from this?** For example, it has come to attention that 1/5 of all active filters discriminate against new users via the `!("confirmed" in user_groups)` pattern. Are there other tendencies of interest?

4. **Is there a qualitative difference between the tasks/patterns of public and hidden filters?** According to the guidelines for filter creation, general filters should be public while filters targeting particular users should be hidden. Is there something more to be learnt from an examination of hidden filters' patterns? Do they actually conform to the guidelines?

5. **How are false positives handled?** Have filters been shut down regularly, because they matched more false positives than they had real value? Are there big amounts of false positives that corrupt the filters hit data and thus the interpretations offered by the current work?

6. **To implement a bot or to implement a filter?** An ethnographic inquiry into if an editor is simultaneously an edit filter manager and a bot operator when faced with a new problem, how do they decide which mechanism to employ for the solution?

7. **What are the repercussions on affected editors?** An ethnographic study of the consequences of edit filters for editors whose edits are filtered. Do they experience frustration or alienation? Do they understand what is going on? Or do they experience for example edit filters' warnings as

helpful and appreciate the hints they have been given and use them to improve their collaboration?

8. **What are the differences between how filters are governed on EN Wikipedia compared to other language versions?** Different Wikipedia language versions each have a local community behind them. These communities vary, sometimes significantly, in their modes of organisation and values. It would be very insightful to explore disparities between filter governance and the types of filters implemented between different language versions.

9. **Are edit filters a suitable mechanism for fighting harassment?** A disturbing rise in online personal attacks and harassment is observed in a variety of online spaces, including Wikipedia [DRS+14]. The Wikimedia Foundation sought to better understand harassment in their projects via a Harassment Survey conducted in 2015 [Wik15]. According to the edit filter noticeboard archives [Wik19z], there have been some attempts to combat harassment by means of filters. The tool is also mentioned repeatedly in the timeline of Wikipedia's Community Health Initiative [Wik19o] which seeks to reduce harassment and disruptive behaviour on Wikipedia. An evaluation of its usefulness and success at this task would be really interesting.

10. **(How) has the notion of "vandalism" on Wikipedia evolved over time?** By comparing older and newer filters, or respectively updates in filter patterns, it could be investigated whether there has been a qualitative change in the interpretation of the "vandalism" notion on Wikipedia.

11. **What are the urgent situations in which edit filter managers are given the freedom to act as they see fit and ignore best practices of filter adoption?** (i.e. switch on a filter in log only mode first and announce it on the notice board so others can have a look)? Who determines they are urgent? These cases should be scrutinised extra carefully since "urgent situations" have historically always been an excuse for cuts in civil liberties.

7. Conclusion

The present thesis conducted an initial inquiry into an important quality control mechanism on Wikipedia previously unexplored by the scientific community—the edit filters. The role of edit filters in Wikipedia's quality control ecosystem, the tasks the filters take care of, as well as some historical trends in filters' usage were studied. It was further discussed why such an old-school rule-based technology is still actively used today when more advanced machine learning approaches exist. Additionally, interesting paths for future research were suggested.

Summing up the most prominent results, edit filters, together with page protection and title/spam blacklist mechanisms, are the first mechanism verifying incoming contributions. By acting on unpublished edits they can disallow unconstructive ones directly and thus reduce the workload for other mechanisms. At the time of their introduction, the need was felt for a mechanism that swiftly prohibited obvious but difficult to remove vandalism, often caused by the same highly motivated malicious users. Although mass-scale page moves to nonsensical names could be taken care of by admin bots, edit filters were viewed as a neater solution since this way such edits are not published at all. Also, with some dissatisfaction with bots' development processes (poorly tested and not available source code, low responsiveness of some bot operators), the opportunity for a clean start with a new tool was taken. Apart from targeting single highly motivated disrupting editors, edit filters take care of "common newbie mistakes" such as publishing text not formatted according to wikisyntax or erasing an entire page instead of properly moving it to a different name, or suggesting it to the formal Articles for Deletion process. By issuing warnings with helpful pointers towards possible alternative actions, edit filters allow a unintentionally disrupting editor to improve their contribution before re-submitting it. With feedback provided immediately at publication, the revert first-ask questions later approach of other mechanisms (which frustrates and alienates good intentioned newcomers [HGMR13]) is inverted. Compared to machine learning techniques, rule-based systems such as the edit filters have the advantage of providing higher amount of control for their operators and being easier to use and understand which also enhances accountability.

Taking a step back, according to the Wikipedian community, people adding made-up information like references to Brazilian aardvarks or proclaiming themselves mayors of small Chinese towns [Wik19g] shall preferably not publish at all. If this type of disruption is to be handled with edit filters, two approaches seem feasible: Warn editors adding the information that their contribution does not contain any references, or outright disallow such edits (which does not solve the problem of freely invented sources), but that was pretty much it. Albeit edit filters may not be the ideal mechanism to deal with hoaxes, what they can do effectively is prevent someone from moving hundreds of pages to titles containing "on wheels" [Wik19av], thus sparing vandal fighters the need to track down and undo these changes, allowing them to use their time more productively by for example fact checking unverified claims and hence reducing the number of fake aardvarks and increasing the overall credibility of the project.

It is impressive how in under 20 years "a bunch of nobodies created the world's greatest encyclopedia" to quote new media researcher Andrew Lih [Lih09]. This was possible, among other things, because there was one Wikipedia to which everybody contributed. As the project and its needs for quality control grew, a lot of processes became more centralised [HGMR13]. It is, at the end, easier to maintain power and control in a centralised infrastructure. However, centralisation facilitates not only the contribution of everyone towards a common goal—creating the world's biggest knowledge database, but also control. It is not an accident that at the very introduction of the AbuseFilter extension, critical voices expressed the concern that a really powerful secret tool was created to which very few people were to have access and thereby a large-scale censorship infrastructure was being installed [Wik19ac]. If there were multiple comparable projects, all of them had to be censored in order to silence people. With Wikipedia being the first go-to source of information for a vast quantity of people all over the world today, the debate whose knowledge is included and who decides what knowledge is worth preserving is essential [Tka14]. In the present moment, it is more relevant than ever: In March 2019, the European Parliament basically voted the introduction of upload filters all over the Internet [Par19]. In a way, that is exactly what Wikipedia's edit filters are—they are triggered prior to publication and are able to effectively disallow upload of undesired content.

Since Wikipedia is distinctly relevant for the shaping of public opinion, despite its "neutral point of view" policy [Wik19au] it is inherently political. At the beginnings of this research, I heard from a former colleague that there was an edit filter on the German Wikipedia targeting gendering. "To gender" is a linguistic praxis whereby words referring to people are explicitly marked to designate more genders than the standardly used generic masculine. It is a political praxis aiming to uncover under-represented groups and their experiences through the conscious use of language. Even though no linguistic norm has established gendering to date, conscious decisions for or against the praxis are political, and so are technologies implementing these decisions. As it turned out, no such filter existed on the German Wikipedia [1]. This illustrates a point though: Artefacts do have politics [Win80] and as Lawrence Lessig puts it, it

[1] Although, during one of the monthly WomenEdit meetups [Wik19bi] hosted at Wikimedia Deutschland office, women active in the German Wikipedia community related that there was a strong general backlash against gendering. The community is also extremely male dominated.

is up to us to decide what values we embed in the systems we create [Les06].

Bibliography

[AH18] Sumit Asthana and Aaron Halfaker. With few eyes, all hoaxes are deep. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):21, 2018. http://delivery.acm.org/10.1145/3280000/3274290/cscw021-asthana.pdf.

[BF14] Sönke Bartling and Sascha Friesike. *Opening science: The evolving guide on how the internet is changing research, collaboration and scholarly publishing*. Springer, 2014.

[CDFN12] Imogen Casebourne, Chris Davies, Michelle Fernandes, and Naomi Norman. Assessing the accuracy and quality of wikipedia entries compared to popular online encyclopaedias: A comparative preliminary study across disciplines in english, spanish and arabic. *Epic, Brighton, UK. Accedido o*, 9(10):2012, 2012.

[Cha06] Kathy Charmaz. *Constructing Grounded Theory*. SAGE, 2006. http://www.sxf.uevora.pt/wp-content/uploads/2013/03/Charmaz_2006.pdf.

[DHN+17] John Danaher, Michael J Hogan, Chris Noone, Rónán Kennedy, Anthony Behan, Aisling De Paor, Heike Felzman, Muki Haklay, Su-Ming Khoo, John Morison, Maria Helen Murphy, Niall O'Brolchain, Burkhard Schafer, and Kalpana Shankar. Algorithmic governance: Developing a research agenda through the power of collective intelligence. *Big Data & Society*, 4, 2017. https://journals.sagepub.com/doi/pdf/10.1177/2053951717726554.

[DRS+14] Maeve Duggan, Lee Rainie, Aaron Smith, Cary Funk, Amanda Lenhart, and Mary Madden. Online harassment. October 2014. https://www.pewinternet.org/2014/10/22/online-harassment/.

[Eld16] Jeff Elder. Inside the game of sports vandalism on Wikipedia, January 2016. Retreived 24 July 2019 from https://blog.wikimedia.org/2016/01/06/sports-vandalism-on-wikipedia/.

[FG12] Heather Ford and R Stuart Geiger. Writing up rather than writing down: Becoming Wikipedia literate. In *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration*, page 16. ACM, 2012. http://www.stuartgeiger.com/writing-up-wikisym.pdf.

[Gar19a] Andrew Garret. AbuseFilter extension source code, 2019. Retreived 3 July 2019 from https://gerrit.wikimedia.org/r/plugins/gitiles/mediawiki/extensions/AbuseFilter/+/refs/heads/master.

[Gar19b] Andrew Garret. AbuseFilter extension tables, 2019. Retreived 9 March 2019 from https://gerrit.wikimedia.org/r/plugins/gitiles/mediawiki/extensions/AbuseFilter/+/refs/heads/master/abusefilter.tables.sql.

[Gei09] R Stuart Geiger. The social roles of bots and assisted editing programs. In *Int. Sym. Wikis*, 2009. http://www.stuartgeiger.com/papers/geiger-wikisym-bots.pdf.

[Gei11] R. Stuart Geiger. The lives of bots. In Geert W. Lovink and Nathaniel Tkacz, editors, *Critical point of view. A Wikipedia Reader*, pages 78–93. Institute of Network Cultures, Amsterdam, 2011. https://www.networkcultures.org/_uploads/%237reader_Wikipedia.pdf.

[Gei14] R Stuart Geiger. Bots, bespoke code and the materiality of software platforms. *Information, Communication & Society*, 17, 2014. http://stuartgeiger.com/bespoke-code-ics.pdf.

[Gei17] R Stuart Geiger. Beyond opening up the black box: Investigating the role of algorithmic systems in Wikipedian organizational culture. *Big Data & Society*, 4, 2017. http://stuartgeiger.com/algoculture-bds.pdf.

[GH13] R Stuart Geiger and Aaron Halfaker. When the levee breaks: without bots, what happens to Wikipedia's quality control processes? In *Proceedings of the 9th International Symposium on Open Collaboration*, page 6. ACM, 2013. http://stuartgeiger.com/wikisym13-cluebot.pdf.

[GH17] R Stuart Geiger and Aaron Halfaker. Operationalizing conflict and cooperation between automated software agents in Wikipedia: A replication and expansion of "Even good bots fight". *unpublished*, 2017. https://upload.wikimedia.org/wikipedia/commons/f/f4/Operationalizing-conflict-bots-wikipedia-cscw-preprint.pdf.

[Gil05] Jim Giles. Internet encyclopaedias go head to head, 2005.

[Git19] Gitlab repository of the thesis, 2019. https://git.imp.fu-berlin.de/luvaseva/wikifilters.

[GR10] R Stuart Geiger and David Ribes. The work of sustaining order in Wikipedia: the banning of a vandal. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pages 117–126. ACM, 2010. http://www.stuartgeiger.com/papers/cscw-sustaining-order-wikipedia.pdf.

[GR11] R Stuart Geiger and David Ribes. Trace ethnography: Following coordination through documentary practices. In *Proceedings of the 44th Hawaii International Conference on System Sciences*, 2011. http://www.stuartgeiger.com/trace-ethnography-hicss-geiger-ribes.pdf.

[Hen13] Casper Henderson. *The Book of Barely Imagined Beings: A 21st Century Bestiary.* University of Chicago Press, 2013. Retreived 21 July 2019 from https://books.google.co.uk/books?id=UQmZN2z92woC&pg=PA10&dq=%22Brazilian+aardvark%22&hl=en&sa=X&ved=0ahUKEwis0dWE4q7RAhULWBoKHYHABkwQ6AEIJTAC#v=onepage&q=%22Brazilian%20aardvark%22&f=false.

[HGMR13] Aaron Halfaker, R Stuart Geiger, Jonathan T Morgan, and John Riedl. The rise and decline of an open collaboration system: How Wikipedia's reaction to popularity is causing its decline. *American Behavioral Scientist*, 57(5):664–688, 2013. https://stuartgeiger.com/papers/abs-rise-and-decline-wikipedia.pdf.

[HKR11] Aaron Halfaker, Aniket Kittur, and John Riedl. 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*, pages 163–172. ACM, 2011. https://www-users.cs.umn.edu/~halfaker/publications/Don't_Bite_the_Newbies/halfaker11bite-personal.pdf.

[HR12] Aaron Halfaker and John Riedl. Bots and cyborgs: Wikipedia's immune system. *Computer*, 45(3):79–82, 2012. http://stuartgeiger.com/bots-cyborgs-halfaker.pdf.

[HS15] Benjamin Mako Hill and Aaron Shaw. Page protection: another missing dimension of wikipedia research. In *Proceedings of the 11th International Symposium on Open Collaboration*, page 15. ACM, 2015. http://delivery.acm.org/10.1145/2790000/2789846/a15-mako-hill.pdf?ip=95.91.215.208&id=2789846&acc=OA&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2EEE82137EC095B955&__acm__=1564037002_dac24f8db14757ad120e77767efc09a8.

[HT15] Aaron Halfaker and Dario Taraborelli. Artificial intelligence service “ORES” gives Wikipedians X-ray specs to see through bad edits, 2015. Retreived 25 March 2019 from https://blog.wikimedia.org/2015/11/30/artificial-intelligence-x-ray-specs/.

[KCP$^+$07] Aniket Kittur, Ed Chi, Bryan A Pendleton, Bongwon Suh, and Todd Mytkowicz. Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. *World wide web*, 1(2):19, 2007. https://www.researchgate.net/profile/Bongwon_Suh/publication/200772541_Power_of_the_Few_vs_Wisdom_of_the_Crowd_Wikipedia_and_the_Rise_of_the_Bourgeoisie/links/53d1e6220cf2a7fbb2e95533.pdf.

[Les06] Lawrence Lessig. *Code version 2.0*. Basic Books, 2006. http://codev2.cc/download+remix/Lessig-Codev2.pdf.

[Lih09] Andrew Lih. *The Wikipedia revolution: How a bunch of nobodies created the world’s greatest encyclopedia*. Hachette Books, 2009.

[Liv16] Randall M Livingstone. Population automation: An interview with Wikipedia bot pioneer Ram-Man. *First Monday*, 21(1), 2016. https://firstmonday.org/ojs/index.php/fm/article/view/6027/5189.

[MBDH13] Claudia Müller-Birn, Leonhard Dobusch, and James D Herbsleb. Work-to-rule: the emergence of algorithmic governance in Wikipedia. In *Proceedings of the 6th International Conference on Communities and Technologies*, pages 80–89. ACM, 2013. http://www.dobusch.net/pub/uni/MuellerBirn-Dobusch-Herbsleb(2013)Work-to-Rule.pdf.

[Med19a] Abuse filter extension actions, 2019. Retreived 13 March 2019 from https://www.mediawiki.org/w/index.php?title=Extension:AbuseFilter/Actions&oldid=3095920.

[Med19b] Abuse filter extension conditions, 2019. Retreived 22 July 2019 from https://www.mediawiki.org/w/index.php?title=Extension:AbuseFilter/Conditions&oldid=3228642.

[Med19c] Abuse filter extension documentation, 2019. Retreived 3 July 2019 from https://www.mediawiki.org/w/index.php?title=Extension:AbuseFilter&oldid=3270512.

[Med19d] Abuse filter extension rules format, 2019. Retreived 3 July 2019 from https://www.mediawiki.org/w/index.php?title=Extension:AbuseFilter/Rules_format&oldid=3240087.

[Med19e] Mediawiki page protection, 2019. Retreived 22 July 2019 from https://www.mediawiki.org/w/index.php?title=Help:Protecting_and_unprotecting_pages&oldid=2981908.

[Med19f] Mediawiki: Spam blacklist, 2019. Retreived 1 July 2019 from https://en.wikipedia.org/w/index.php?title=MediaWiki:Spam-blacklist&oldid=904319854.

[Med19g] Mediawiki: Title blacklist, 2019. Retreived 1 July 2019 from https://en.wikipedia.org/w/index.php?title=MediaWiki:Titleblacklist&oldid=904314604.

[Mes19] Meshvogel. AbuseFilter patch to database replication scripts renewing the public dump of abuse_filter_history table, 2019. Retreived 19 July 2019 from https://gerrit.wikimedia.org/r/#/c/operations/puppet/+/498773/.

[ORE19] ORES homepage, 2019. Retreived 16 July 2019 from https://ores.wmflabs.org/.

[Par19] European Parliament. European Parliament adopts Directive on Copyright in the Digital Single Market, 2019. Retreived 24 July 2019 from https://www.europarl.europa.eu/doceo/document/TA-8-2019-0231_EN.html.

[Pla16a] Phabricator Collaboration Platform. AbuseFilter extension issues created in the period May 2015–May 2016, 2016. Retreived 20 July 2019 from https://phabricator.wikimedia.org/project/board/217/query/T4UBDo9V4u1n/.

[Pla16b] Phabricator Collaboration Platform. Bring back abuse_filter_history view, 2016. Retreived 9 March 2019 from https://phabricator.wikimedia.org/T123978.

[PSG08] Martin Potthast, Benno Stein, and Robert Gerling. Automatic vandalism detection in Wikipedia. In *ECIR*, 2008. https://webis.de/downloads/publications/papers/stein_2008c.pdf.

[pyw] pywikibot: A python library and collection of scripts that automate work on mediawiki sites. Retreived 20 July 2019 from https://doc.wikimedia.org/pywikibot/master/.

[Ran14] Eric Randall. How a racoon became an aardvark. *The New Yorker*, May 2014. Retreived on 21 July 2019 from https://www.newyorker.com/tech/annals-of-technology/how-a-raccoon-became-an-aardvark.

[Ray99] Eric Raymond. The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3):23–49, 1999. http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html.

[Sig09] Signpost. Abuse Filter is enabled, 2009. Retreived 25 February 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Wikipedia_Signpost/2009-03-23/Abuse_Filter&oldid=878994386.

[Ste01] Steve Stemler. An overview of content analysis. *Practical assessment, research & evaluation*, 7(17):137–146, 2001.

[Tka14] Nathaniel Tkacz. *Wikipedia and the Politics of Openness*. University of Chicago Press, 2014.

[WCV+11] Andrew G West, Jian Chang, Krishna Venkatasubramanian, Oleg Sokolsky, and Insup Lee. Link spamming wikipedia for profit. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, pages 152–161. ACM, 2011. https://repository.upenn.edu/cgi/viewcontent.cgi?article=1508&context=cis_papers.

[Wik08] Wikipedia: Coati becomes a brazilian aardvark, 2008. Retreived 21 July 2019 from https://en.wikipedia.org/w/index.php?title=Coati&oldid=225140818.

[Wik14] Wikipedia: Media reference to the coati being called a brazilian aardvark, 2014. Retreived 21 July 2019 from https://en.wikipedia.org/w/index.php?title=Coati&oldid=607642706.

[Wik15] Wikimedia foundation: Harassment survey, 2015. Retreived 24 July 2019 from https://upload.wikimedia.org/wikipedia/commons/5/52/Harassment_Survey_2015_-_Results_Report.pdf.

[Wik19a] Wikimedia foundation: Mission, 2019. Retreived 24 July 2019 from https://wikimediafoundation.org/about/mission/.

[Wik19b] Wikimedia foundation: Projects, 2019. Retreived 24 July 2019 from https://wikimediafoundation.org/our-work/wikimedia-projects/.

[Wik19c] Wikimedia: Toolforge, 2019. Retreived 20 July 2019 from https://wikitech.wikimedia.org/w/index.php?title=About_Toolforge&oldid=1826480.

[Wik19d] Wikipedia: About, 2019. Retreived 20 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:About&oldid=891256910.

[Wik19e] Wikipedia: Abuselog, 2019. Retreived 12 July 2019 from https://en.wikipedia.org/wiki/Special:AbuseLog.

[Wik19f] Wikipedia: Administrator intervention against vandalism, 2019. Retreived 11 April 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Administrator_intervention_against_vandalism&oldid=891917401.

[Wik19g] Wikipedia: Administrators noticeboard—chen fang hoax, 2019. Retreived 24 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Administrators%27_noticeboard/Archive241&oldid=891675599#Fictional_entry?

[Wik19h] Wikipedia: Antivandalbot, 2019. Retreived 16 July 2019 from https://en.wikipedia.org/w/index.php?title=User:AntiVandalBot&oldid=647615013.

[Wik19i] Wikipedia: Article feedback v5, 2019. Retreived 3 July 2019 from https://www.mediawiki.org/w/index.php?title=Extension:ArticleFeedbackv5&oldid=3136840.

[Wik19j] Wikipedia: Articles for deletion, 2019. Retreived July 25 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Articles_for_deletion&oldid=892360111.

[Wik19k] Wikipedia: Assume good faith, 2019. Retreived 26 March 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Assume_good_faith&oldid=889253693.

[Wik19l] Wikipedia: Bot operators category, 2019. Retreived 5 July 2019 from https://en.wikipedia.org/w/index.php?title=Category:Wikipedia_bot_operators&oldid=833970789.

[Wik19m] Wikipedia: Cluebot, 2019. Retreived 16 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Bots/Requests_for_approval/ClueBot&oldid=891685765.

[Wik19n] Wikipedia: Cluebot ng, 2019. Retreived 16 July 2019 from https://en.wikipedia.org/w/index.php?title=User:ClueBot_NG&oldid=391868393.

[Wik19o] Wikipedia: Community health initiative, 2019. Retreived 24 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Community_health_initiative&oldid=905253115.

[Wik19p] Wikipedia: Datbot, 2019. Retreived 12 June 2019 from https://en.wikipedia.org/w/index.php?title=User:DatBot&oldid=900858894.

[Wik19q] Wikipedia: Dumbbot, 2019. Retreived 16 July 2019 from https://en.wikipedia.org/w/index.php?title=User:DumbBOT&oldid=794414317.

[Wik19r] Wikipedia: Edit filter, 2019. Retreived 25 February 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Edit_filter&oldid=877829572.

[Wik19s] Wikipedia: Edit filter 365 detailed page, 2019. Retreived 12 July 2019 from https://en.wikipedia.org/wiki/Special:AbuseFilter/365.

[Wik19t] German wikipedia: Edit filter, 2019. Retreived 17 July 2019 from https://de.wikipedia.org/w/index.php?title=Wikipedia:Bearbeitungsfilter&oldid=188476517.

[Wik19u] Wikipedia: Edit filter documentation, 2019. Retreived 25 April 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Edit_filter/Documentation&oldid=879787821.

[Wik19v] Spanish wikipedia: Edit filter, 2019. Retreived 17 July 2019 from https://es.wikipedia.org/w/index.php?title=Wikipedia:Filtro_de_ediciones&oldid=104083930.

[Wik19w] Wikipedia: Edit filter helper, 2019. Retreived 17 March 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Edit_filter_helper&oldid=878127027.

[Wik19x] Wikipedia: Edit filter managers list, 2019. Retreived 10 May 2019 from https://en.wikipedia.org/wiki/Special:ListUsers/abusefilter.

[Wik19y] Wikipedia: Edit filter noticeboard, 2019. Retreived 12 March 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Edit_filter_noticeboard&oldid=887086700.

[Wik19z] Wikipedia: Edit filter used for combating harassment, 2019. Retreived 6 June 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Edit_filter_noticeboard/Archive_2&oldid=803551337#Exploring_how_the_Edit_filter_can_be_used_to_combat_harassment.

[Wik19aa] Wikipedia: Edit filter report false positives, 2019. Retreived 13 March 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Edit_filter/False_positives&oldid=879367604.

[Wik19ab] Russian wikipedia: Edit filter, 2019. Retreived 17 July 2019 from https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%A4%D0%B8%D0%BB%D1%8C%D1%82%D1%80_%D0%BF%D1%80%D0%B0%D0%B2%D0%BE%D0%BA&oldid=100429471.

[Wik19ac] Wikipedia: Edit filter talk archive 1, 2019. Retreived 22 May 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia_talk:Edit_filter/Archive_1&oldid=884572675.

[Wik19ad] Wikipedia: Edit filter talk archive 1 clarification, 2019. Retreived 22 May 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia_talk:Edit_filter/Archive_1&oldid=884572675#Clarification.

[Wik19ae] Wikipedia: Edit filter talk archive for 2015–2016, 2019. Retreived 20 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia_talk:Edit_filter/Archive_7&oldid=901909123.

[Wik19af] Wikipedia: Edit filter talk archive name change, 2019. Retreived 25 April 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia_talk:Edit_filter/Archive_3&oldid=883700704#Request_for_name_change.

[Wik19ag] Wikipedia: Emausbot, 2019. Retreived 16 July 2019 from https://en.wikipedia.org/w/index.php?title=User:EmausBot&oldid=876175163.

[Wik19ah] Wikipedia: Faq why might a category list not be up to date, 2019. Retreived 5 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:FAQ/Categorization&oldid=887018121#Why_might_a_category_list_not_be_up_to_date?

[Wik19ai] Wikipedia: Harassment, 2019. Retreived 5 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Harassment&oldid=886343748.

[Wik19aj] Wikipedia: Hbc aiv helperbot, 2019. Retreived 16 July 2019 from https://en.wikipedia.org/w/index.php?title=User:HBC_AIV_helperbot&oldid=794414856.

[Wik19ak] Wikipedia: Huggle tool, 2019. Retreived 7 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Huggle&oldid=890902601.

[Wik19al] Wikipedia: Instructions for introducing a new edit filter, 2019. Retreived 12 March 2019 from https:

//en.wikipedia.org/w/index.php?title=Wikipedia:Edit_filter/Instructions&oldid=844579470.

[Wik19am] Wikipedia: List of administrators, 2019. Retreived 9 March 2019 from https://en.wikipedia.org/w/index.php?title=Special:ListUsers/sysop.

[Wik19an] Wikipedia: Long term abuse, 2019. Retreived 7 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Long-term_abuse&oldid=863668947.

[Wik19ao] Wikipedia: Lupin's anti-vandal tool, 2019. Retreived 7 July 2019 from https://en.wikipedia.org/w/index.php?title=User:Lupin/Anti-vandal_tool&oldid=892557860.

[Wik19ap] Wikipedia: Martinbot, 2019. Retreived 16 July 2019 from https://en.wikipedia.org/w/index.php?title=User:MartinBot&oldid=873666125.

[Wik19aq] Wikipedia: Mr.z-bot, 2019. Retreived 7 July 2019 from https://en.wikipedia.org/w/index.php?title=User:Mr.Z-bot&oldid=898492130.

[Wik19ar] Wikipedia: Musikbot abusefilterirc task, 2019. Retreived 3 July 2019 from https://en.wikipedia.org/w/index.php?title=User:MusikBot/AbuseFilterIRC&oldid=885138803.

[Wik19as] Wikipedia: Musikbot filtermonitor task, 2019. Retreived 3 July 2019 from https://en.wikipedia.org/w/index.php?title=User:MusikBot/FilterMonitor&oldid=748920629.

[Wik19at] Wikipedia: Musikbot stalefilters task, 2019. Retreived 3 July 2019 from https://en.wikipedia.org/w/index.php?title=User:MusikBot/StaleFilters&oldid=763698371.

[Wik19au] Wikipedia: Neutral point of view, 2019. Retreived 23 July, 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Core_content_policies&oldid=903666779.

[Wik19av] Wikipedia: On wheels vandalism, 2019. Retreived 25 July, 2019 from https://en.wikipedia.org/w/index.php?limit=50&title=Special%3AContributions&contribs=user&target=Willy+on+wheels+for+President+2008&namespace=&tagfilter=&start=&end=.

[Wik19aw] Wikipedia: Page protection, 2019. Retreived 11 April 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Requests_for_page_protection&oldid=891912507.

[Wik19ax] Wikipedia: Privacy of general vandalism filters, 2019. Retrieved 17 July 2019 from https://en.wikipedia.org/w/index.php?oldid=784131724#Privacy_of_general_vandalism_filters.

[Wik19ay] Wikipedia: Requesting new edit filters, 2019. Retrieved 12 March 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Edit_filter/Requested&oldid=871023624.

[Wik19az] Wikipedia: Rollback, 2019. Retrieved 5 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Rollback&oldid=901761637.

[Wik19ba] Wikipedia: Sockpuppetry, 2019. Retrieved 5 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Sock_puppetry&oldid=903464918.

[Wik19bb] Wikipedia: Stiki tool, 2019. Retrieved 31 March 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:STiki&oldid=879253675.

[Wik19bc] Wikipedia: Text table, 2019. Retrieved 20 July 2019 from https://www.mediawiki.org/w/index.php?title=Manual:Text_table&oldid=3287673.

[Wik19bd] Wikipedia: Twinkle tool, 2019. Retrieved 7 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Twinkle&oldid=902956777.

[Wik19be] Wikipedia: Usernames for administrator attention, 2019. Retreived 12 July 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Usernames_for_administrator_attention&oldid=905957698.

[Wik19bf] Wikipedia: User:werdna, 2019. Retrieved 24 July 2019 from https://en.wikipedia.org/w/index.php?title=User:Werdna&oldid=661901764.

[Wik19bg] Wikipedia: Vandalism, 2019. Retrieved 26 March 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Vandalism&oldid=886784937.

[Wik19bh] Wikipedia: Vandalism types, 2019. Retrieved 27 June 2019 from https://en.wikipedia.org/w/index.php?title=Wikipedia:Vandalism_types&oldid=876716354.

[Wik19bi] Wikipedia: Women edit, 2019. Retrieved 25 July 2019 from https://de.wikipedia.org/w/index.php?title=Wikipedia:WomenEdit&oldid=190243967.

[Wik19bj] Wikipedia: Xlinkbot, 2019. Retreived 16 July 2019 from https://en.wikipedia.org/w/index.php?title=User:XLinkBot&oldid=863906174.

[Win80] Langdon Winner. Do artifacts have politics? *Daedalus*, pages 121–136, 1980. https://www.jstor.org/stable/pdf/20024652.pdf.

[WKL10] Andrew G West, Sampath Kannan, and Insup Lee. Stiki: an anti-vandalism tool for Wikipedia using spatio-temporal analysis of revision metadata. In *Proceedings of the 6th International Symposium on Wikis and Open Collaboration*, page 32. ACM, 2010. https://repository.upenn.edu/cgi/viewcontent.cgi?article=1490&context=cis_papers.

Appendices

A. Code Book

This section provides a detailed overview of all the codes[2] used for the manual tagging of edit filters. The purpose of the coding was to gain insight into the specific tasks filters are applied for on English Wikipedia.

Vandalism	
Structure related	
avoidant_vandalism	removing tags or other content in order to avoid that own edits are deleted or reverted, or other sanctions Example: 419 “User removing himself from AIV”
image_vandalism	From Wikipedia’s Vandalism Typology: “Uploading shock images that do not belong at all on Wikipedia; Inappropriately placing explicit images legitimately used on Wikipedia on pages where they do not belong” [Wik19bh] Example: 131 “Removal of controversial images”
link_vandalism	Partially adopted from Wikipedia Vandalism Typology: “adding external links to non-notable or irrelevant sites; adding external links that may belong on another Wikipedia page, but have no relevance to the subject matter of the page to which they are added” [Wik19bh] Example: 24 “Sneaky link vandalism”
page_move_vandalism	moving a page (i.e. renaming the page), mostly to some nonsensical name Example: 883 “Page moves to bad words or other vandalism”
talk_page_vandalism	malicious editing of talk pages: e.g. modifiyng or removing other users’ comments from discussions Example: 420 “Large removal of talk page content by IP”

[2] Here, I use the words “codes”, “tags” and “labels” interchangeably.

template_vandalism	From Wikipedia's Vandalism Typology: "Modifying a template in a harmful or disruptive manner. This is especially serious, because it'll negatively impact the appearance of multiple pages. Some templates appear on hundreds of pages." [Wik19bh] Example: 740 "Template hijacking"
username_vandalism	creating accounts with offensive or disruptive usernames Example: 827 "Abusive username activity"
Content related	
hoaxing	deliberately inserting false information or using non-existent references Example: 686 "IP adding possibly unreferenced material to BLP"
profanity_vandalism	inserting profanities into articles in general, without them being targeted at a person (the latter is covered by "personal_attacks") Example: 380 "Multiple obscenities"
silly_vandalism	blatant, immediately obvious vandalism, such as inserting repeating random characters or other intentional nonsense Example: 135 "Repeating characters"
trolling	seeking to disrupt productive work, e.g. by starting off-topic discussions and trying to provoke emotional responses; assigned when "trolling" is explicitely referenced in the filter name; Example: 896 "ANI trolling"
Ideologically motivated	
politically_motivated	disrupting on explicitely politic matters Example: 119 "Macedonia naming conflict"
religiously_motivated	disrupting on topics related to religion Example: 710 "Muhammad vandal"
Spam/malware/etc.	
malware	"malware" is explicitely mentioned in the filter's name Example: 243 "WikiMedia Viewer possible malware"
phishing	"phishing" is explicitely mentioned in the filter's name Example: 870 "nowiki phishing"

spam	inserting links to promotional content regardless whether they are related to the page being edited or not Example: 862 “Arabic string spam”
General vandalism	
bot_vandalism	vandalism caused by an automated agent Example: 425 “Magic/astrology spambots”
general_vandalism	vandalism for which none of the more specific tags applied Example: 194 “Michael Jackson new page vandalism”
Hardcore vandalism (the really malicious cases)	
doxxing	disclosing private information of other people (e.g. address, contact details, details about their life not known to the public) without their consent; often with the purpose of facilitating organised harassment Example: 76 “Adding email address”
harassment	Defined by Wikipedia as “stop[ping] other editors from enjoying Wikipedia by making threats, repeated annoying and unwanted contacts, repeated personal attacks, intimidation” [Wik19ai]; assigned when filter contains “harassment” in its name/comments; Example: 792 “Harassments”
hidden_vandalism	assigned to hidden filters where a more specific tag could not be determined Example: 13 “Knave vandalism”
impersonation	trying to pose as another editor; mostly assgined when “impersonation” metioned in the filter’s name/comments; Example: 568 “SPI Clerk impersonation”
long_term_abuse	Defined by Wikipedia as “The user has been abusing Wikipedia over a long duration of time. The user account has a history of repeated egregious disruption, and despite indefinite block or ban, continues vandalism and/or abuse beyond the point of any usual blocked user.” [Wik19an]; assigned when filter has “Long term abuse”, “LTA”, or similar in its name; Example: 51 “LTA Username / LTA IP hopping disruption (Oshwah)”

not_polite	interacting in a non-civil manner, without being directly a personal attack (e.g. “shouting”) Example: 521 “Feedback: All caps”
personal_attacks	directly insulting particular persons (be it other editors or persons who are the subject matter of an article) Example: 294 “Personal attacks”
sockpuppetry	using multiple accounts to “mislead, deceive, vandalize or disrupt; to create the illusion of greater support for a position; to stir up controversy; or to circumvent a block, ban, or sanction” [Wik19ba]; assigned mostly to filters containing “sock”, “sockpuppets”, or similar in their name; sockpuppetry is often long term abuse, but not necessarily all long term abuse involves sock puppetry Example: 16 “Prolific socker I”
Good faith	
Policy violations	
bad_style	deviating from what is percieved a good encyclopedic style 65 “Excessive whitespace”
copyright_violation	adding content which potentially violates copyright: e.g. images without license information Example: 798 “Possible copyvio for image upload”
edit_warring	engaging in edit or respectively revert wars Example: 622 “Genre edit-warring”
wiki_policy	violating Wikipedia’s policies 613 “Signing in article”
Point of view problems	
conflict_of_interest	editing articles about organisations one is affilitated to or receives money from Example: 302 “Possible COI”
self_promotion	editing articles about oneself Example: 214 “Creating articles with title contained in username”
Structure related The tags in this section are somewhat self-explainatory. They designate edits or other editors’ actions that are presumably good intended but still disruptive. The names of the following tags reflect the areas the edit was happening in.	
good_faith	general good faith structure label, assigned when no more specific label could be determined

	Example: 479 “Adding example.jpg to article space”
good_faith_article_creation	Example: 98 “Creating very short new article”
good_faith_categories	Example: 192 “Direct use of stub categories in articles”
good_faith_deletion	Example: 3 “New user blanking articles”
good_faith_edits	Example: 197 “Duplicate section”
good_faith_edit_summary	Example: 703 “Edit summary only consists of article title”
good_faith_external_resources	Example: 220 “Adding external images/links”
good_faith_html	Example: 144 “Hiding content of pages”
good_faith_image	Example: 280 “New user altering images”
good_faith_move	Example: 5 “User self-renaming or moving user talk pages into article talk space”
good_faith_orthography	Example: 432 “Starting new line with lowercase letters”
good_faith_redirect	Example: 35 “New user changing a redirect”
good_faith_refs	Example: 61 “New user removing references”
good_faith_revert	Example: 249 “New user conducting large scale reverts”
good_faith_template	Example: 59 “New user removing templates on image description”
good_faith_test_edits	Example: 18 “Test type edits from clicking on edit bar”
good_faith_userpage	Example: 733 “New user creating a page in someone else’s userspace”
good_faith_wiki_links	

	Example: 753 “wikilinks removed by a new user or IP”
good_faith_wiki_syntax	Example: 26 “Large unwikified additions by new user”
Maintenance	
bug	software bugs from MediaWiki, browser extensions, etc which sometimes cause eroneous syntax Example: 577 “VisualEditor bugs: Strange icons”
general_maintenance	taking care of other maintenance tasks Example: 270 “Interwiki link removal”
general_tracking	tracking certain behaviour in order to determine whether it occurs frequently and how problematic it is Example: 155 “Adding links to youtube”
test	testing patterns to be incorporated in other filters (the filters can be of single edit filter managers or jointly used) Example: 358 “Od Mishehu’s test filter”
Unknown	
unclear	an auxiliary tag for filters that didn’t fit in any other category Example: 642 “OTRS template added by non-OTRS member (global)”

Table A.1.: Code book

B. Extra Figures and Tables

```
abuse_filter
+--------------------+---------------------+------+-----+---------+----------------+
| Field              | Type                | Null | Key | Default | Extra          |
+--------------------+---------------------+------+-----+---------+----------------+
| af_id              | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |
| af_pattern         | blob                | NO   |     | NULL    |                |
| af_user            | bigint(20) unsigned | NO   | MUL | NULL    |                |
| af_user_text       | varbinary(255)      | NO   |     | NULL    |                |
| af_timestamp       | binary(14)          | NO   |     | NULL    |                |
| af_enabled         | tinyint(1)          | NO   |     | 1       |                |
| af_comments        | blob                | YES  |     | NULL    |                |
| af_public_comments | tinyblob            | YES  |     | NULL    |                |
| af_hidden          | tinyint(1)          | NO   |     | 0       |                |
| af_hit_count       | bigint(20)          | NO   |     | 0       |                |
| af_throttled       | tinyint(1)          | NO   |     | 0       |                |
| af_deleted         | tinyint(1)          | NO   |     | 0       |                |
| af_actions         | varbinary(255)      | NO   |     |         |                |
| af_global          | tinyint(1)          | NO   |     | 0       |                |
| af_group           | varbinary(64)       | NO   | MUL | default |                |
+--------------------+---------------------+------+-----+---------+----------------+
```

Figure B.1.: abuse_filter schema

```
abuse_filter_log
+------------------+---------------------+------+-----+---------+----------------+
| Field            | Type                | Null | Key | Default | Extra          |
+------------------+---------------------+------+-----+---------+----------------+
| afl_id           | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |
| afl_filter       | varbinary(64)       | NO   | MUL | NULL    |                |
| afl_user         | bigint(20) unsigned | NO   | MUL | NULL    |                |
| afl_user_text    | varbinary(255)      | NO   |     | NULL    |                |
| afl_ip           | varbinary(255)      | NO   | MUL | NULL    |                |
| afl_action       | varbinary(255)      | NO   |     | NULL    |                |
| afl_actions      | varbinary(255)      | NO   |     | NULL    |                |
| afl_var_dump     | blob                | NO   |     | NULL    |                |
| afl_timestamp    | binary(14)          | NO   | MUL | NULL    |                |
| afl_namespace    | tinyint(4)          | NO   | MUL | NULL    |                |
| afl_title        | varbinary(255)      | NO   |     | NULL    |                |
| afl_wiki         | varbinary(64)       | YES  | MUL | NULL    |                |
| afl_deleted      | tinyint(1)          | NO   |     | 0       |                |
| afl_patrolled_by | int(10) unsigned    | YES  |     | NULL    |                |
| afl_rev_id       | int(10) unsigned    | YES  | MUL | NULL    |                |
| afl_log_id       | int(10) unsigned    | YES  | MUL | NULL    |                |
+------------------+---------------------+------+-----+---------+----------------+
```

Figure B.2.: abuse_filter_log schema

```
abuse_filter_history
+---------------------+---------------------+------+-----+---------+----------------+
| Field               | Type                | Null | Key | Default | Extra          |
+---------------------+---------------------+------+-----+---------+----------------+
| afh_id              | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |
| afh_filter          | bigint(20) unsigned | NO   | MUL | NULL    |                |
| afh_user            | bigint(20) unsigned | NO   | MUL | NULL    |                |
| afh_user_text       | varbinary(255)      | NO   | MUL | NULL    |                |
| afh_timestamp       | binary(14)          | NO   | MUL | NULL    |                |
| afh_pattern         | blob                | NO   |     | NULL    |                |
| afh_comments        | blob                | NO   |     | NULL    |                |
| afh_flags           | tinyblob            | NO   |     | NULL    |                |
| afh_public_comments | tinyblob            | YES  |     | NULL    |                |
| afh_actions         | blob                | YES  |     | NULL    |                |
| afh_deleted         | tinyint(1)          | NO   |     | 0       |                |
| afh_changed_fields  | varbinary(255)      | NO   |     |         |                |
| afh_group           | varbinary(64)       | YES  |     | NULL    |                |
+---------------------+---------------------+------+-----+---------+----------------+
```

Figure B.3.: abuse_filter_history schema

```
abuse_filter_action
+-----------------+---------------------+------+-----+---------+-------+
| Field           | Type                | Null | Key | Default | Extra |
+-----------------+---------------------+------+-----+---------+-------+
| afa_filter      | bigint(20) unsigned | NO   | PRI | NULL    |       |
| afa_consequence | varbinary(255)      | NO   | PRI | NULL    |       |
| afa_parameters  | tinyblob            | NO   |     | NULL    |       |
+-----------------+---------------------+------+-----+---------+-------+
```

Figure B.4.: abuse_filter_action schema