ModSandbox: Rapid Verification of Automated Moderation Rules in an Online Community

ANONYMOUS AUTHOR(S)*

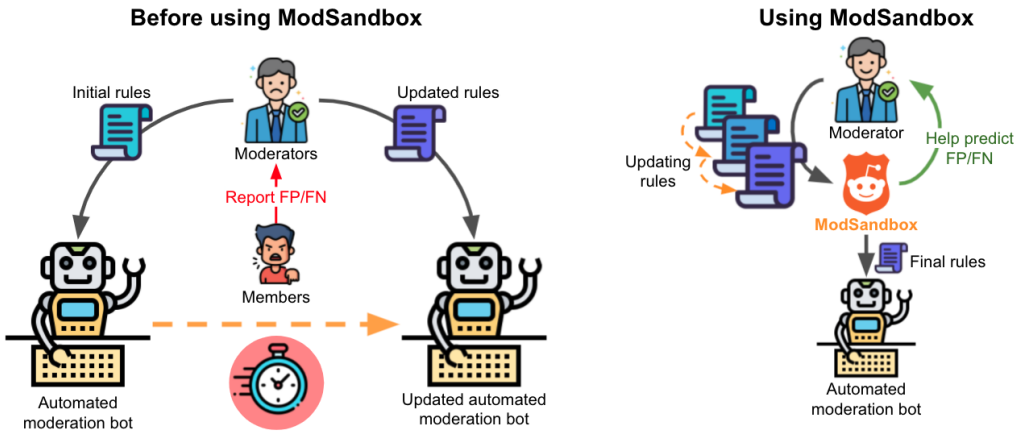

Fig. 1. ModSandbox supports online content moderators with rapid verification and update of their automated moderation rules. It provides features to predict possible false positive and false negatives in real-time and to debug the rules when they cause issues.

Despite the common use of automated bots for online content moderation, human moderators still spend a lot of time monitoring and updating them to ensure they are working as intended. We used Reddit as a case study and asked 102 moderators about their experience using AutoModerator, Reddit's built-in automated bot. We learned that moderators feel frustrated by the lack of testing or debugging tools for AutoModerator. Furthermore, we conducted interviews with six moderators—including the creator of AutoModerator—to understand their moderation processes. Guided by the findings, we built ModSandbox, a novel sandbox system to support the rapid verification of automated rules through false positive/negative prediction and debugging support. We explore the effectiveness of ModSandbox through a user study with ten active moderators. Results show that ModSandbox can help them rapidly test automated rules without changing or affecting their actual community and reduce the total verification time from weeks to hours.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**; **Human computer interaction (HCI)**.

Additional Key Words and Phrases: sociotechnical systems; moderation; automated moderation bots; online communities; virtual sandbox; human-AI collaboration

ACM Reference Format:

Anonymous Author(s). 2022. ModSandbox: Rapid Verification of Automated Moderation Rules in an Online Community . In *Companion Publication of the 2022 Conference on Computer Supported Cooperative Work and Social Computing, November 12–16, 2022, Taipei, Taiwan.* ACM, New York, NY, USA, 31 pages. https://doi.org/10.1145/1122445.1122456

Manuscript submitted to ACM

1 INTRODUCTION

Content moderation on community-based platforms such as Reddit, Discord, and Facebook Groups is often done by a small number of voluntary moderators. Moderators need to monitor the quality of user-created content to take action and manage their communities as desired [27, 49]. However, as the community grows in size and the number of content increases, it becomes almost impossible for the moderators to review all the activities happening in the community, especially when most moderators serve voluntarily in their spare time [18]. For this reason, automated moderation bots are commonly used to perform repetitive moderation tasks.

Even though automated bots are used, many moderators still spend much time manually monitoring whether their automated rules cause false positives (i.e., innocuous contents are removed) or false negatives (i.e., unfavorable contents survive) [8]. This is because a flawed automated rule can negatively impact the community at scale, and a delayed correction of the rule may not recover the damage already inflicted on the community. For example, r/technology, one of Reddit's subcommunities with 4.9 million members, accidentally set an automated rule to remove every post with the word "Tesla", which ended up annoying and losing a large fraction of their community members [16]. Therefore, there are clear needs for testing tools that can preemptively check the effect of automated rules.

This study aims to gain insights into designing a testing system for automated moderation rules. We conducted two rounds of surveys (N=102) and a round of in-depth interviews (N=6) with human moderators on Reddit who have experience configuring AutoModerator—Reddit's built-in automated bot. In the first survey, we asked what makes it difficult to learn or use AutoModerator and found out that having no way to test AutoModerator before deployment is frustrating, making moderators wary of making mistakes. This seemed to hinder the moderators from building complex and precise AutoModerator rules or fixing and updating existing rules. However, they do not have satisfactory testing or debugging methods before deployment. These results led us to envision *a virtual sandbox system* that enables moderators to check their automated rules on real posts without affecting or changing their communities.

In the second survey, we further examined moderators' challenges by focusing on their experience encountering false positives with keyword-based filtering using AutoModerator. The impact of false positives is usually more problematic than that of false negatives [17, 19]. We found that several factors, such as the intricacy of the rules, make it difficult for moderators to identify which part of the rule is causing the problem. These results led us to design specific features for our virtual sandbox system, which helps moderators quickly and easily find false positive and false negative cases and debug their automated rules accordingly.

Through interviews with six Reddit moderators, including the creator of AutoModerator, we found that moderators generally follow an iterative cycle to configure their automated moderation rules. The cycle consists of the following four steps: 1) be motivated to create an automated rule, 2) examine and discuss candidate rules either alone or with other moderators, 3) deploy an AutoModerator rule, and 4) evaluate and update the rule by monitoring the community or receiving user feedback. Steps 2 to 4 are repeated until false positives and false negatives drop below a certain level. Critically, we found that this iterative process takes days or even weeks, especially because Step 4 takes a long time until enough false positive and false negative cases are observed to evaluate and update the rule.

Applying the sandbox system idea to comply with the four-step moderation cycle, we built ModSandbox, a novel sandbox system to support online content moderators with rapid verification of their automated rules. With ModSandbox, moderators can test their rules without harming their actual communities and quickly recognize the presence of any issues since ModSandbox provides them with posts that are likely to be false positives and false negatives. ModSandbox also provides a separated space where they can collect and test against any set of posts, which helps moderators evaluate

their rule by checking its impact on the collected posts. Furthermore, ModSandbox allows moderators to break down and analyze their complex rules, thereby helping them debug the problematic part of the rules.

To evaluate the effectiveness of ModSandbox, we conducted a user study with ten active moderators. The results show that with ModSandbox, moderators rapidly verified and improved their automated rules for keyword-based post-filtering tasks. We also observed that their automated rules have evolved and become more complex to precisely address the given moderation tasks throughout the user study. This is especially useful for community-driven platforms because each community may want to fine-tune automated rules to better fit their own norms [9]. Altogether, ModSandbox provides a solution to reduce moderators' time and effort spent on the moderation cycle from weeks to hours and to enable deploying rules that execute actions as they intended. We conclude our work by discussing the potential of how ModSandbox could be used to assist collaboration between moderators and how it can lower the barrier to entry for novice moderators.

2 RELATED WORK

This section reviews prior studies on content moderation in online communities and automated moderation tools. We also review studies on systems that help users test and debug algorithmic models, which have similar motivations to our work, but in different contexts and applications.

2.1 Centralized or Voluntary User-driven Content Moderation in Online Communities

Content moderation is critical for social platforms because hateful and inappropriate content can make users feel uncomfortable [39] and make them leave their platform entirely [28]. These platforms generally employ one of the two moderation models: 1) centralized corporate models and 2) user-driven models [41]. Social media platforms like Facebook, Instagram, and YouTube have centralized corporate models, employing paid workers to remove harmful content based on their internal rules, reports by users, or found by their filtering algorithms [7, 36, 38]. On the other hand, community-driven sites like Twitch, Reddit, and Wikipedia ("All Wikipedia users are moderators [48]"), which have multiple subcommunities created by users, take user-driven moderation models [32]. They rely on a small group of users called moderators to set rules and moderate the contents in the community.

Centralized content moderation focuses on platform-level "policing" of harmful behaviors such as spreading fake news [31], spreading hate speech [4], sharing unhealthy tags [7], sharing violent or sexual content, or using slurs and swear words. However, while platform-level of centralized moderation effectively regulates harmful behaviors and enhancing brand reputation, they lack incorporating local and cultural context into their moderation decisions [41].

On the other hand, voluntary user-driven content moderation can handle local context into their decisions, providing more freedom to the communities themselves on how they want to flourish their culture and establish norms [40, 50]. Chandrasekharan et al. [9] found micro norms that apply to specific communities in Reddit. For example, one subreddit enforces their members only to use high school-level science to explain new scientific discoveries (e.g., r/AskScience).

In this study, we focus on supporting user-driven content moderation where moderators need to configure their own automated moderation rules to enforce their cultural etiquette or to regulate community-specific attacks or trolling. This moderation model can easily burden voluntary and often non-expert moderators because of its time-consuming and mentally demanding nature. Our approach in designing ModSandbox aims to support even non-expert moderators to easily configure and verify automated moderation rules through features including false positive and false negative prediction and debugging support.

2.2 Automated Content Moderation for Online Community

Recently, researchers have investigated how automated content moderation tools are used in user-driven models [22, 42, 49]. Many of the studies deal with Reddit AutoModerator, one of the most popular and successful automated moderation tools [1, 8, 18, 23, 26, 43]. Kiene et al. [26] reported that one Reddit community, r/NoSleep, handled large influxes of newcomers using AutoModerator. They found that AutoModerator was used to find and remove inappropriate content quickly. Seering et al. [43] interviewed 56 moderators on Twitch, Reddit, and Facebook Group to describe how they contribute to the community with algorithmic tools for dealing with community users' misbehaviors. The authors also found that AutoModerator often mistakenly removes innocuous posts, so the moderator needs to check removed posts manually. Jhaver et al. [18] conducted interviews with 16 Reddit moderators to understand how they use AutoModerator. They report that using AutoModerator created additional tasks for moderators because they had to update it and correct false positives and negatives regularly. Chandrasekharan et al. [8] developed an AI-backed moderation system named Crossmod, an ensemble of classifiers trained by a large number of cross-community posts that are manually removed.

Other studies investigate how automated content moderation is done in Wikipedia, Twitter, and Twitch. In Wikipedia, individual users can moderate content by themselves using third-party automated moderation tools developed by users. Geiger and Ribes [14] first introduced how Wikipedia users fight against vandalism using automated editing tools. In a further study, Geiger and Halfaker [13] attested automated bots' effectiveness of maintaining qualities of Wikipedia articles by analyzing the data collected when one bot went down. Users in Twitter also use automated moderation bots to moderate their own followers and messages. Geiger [12] analyzed a bot-based collective blocklist in Twitter created by voluntary Twitter users to block the harassing accounts. A recent study presented rich information about the blocklist and probed how the stakeholders deal with it through interviews [21]. Voluntary moderators in Twitch use chat-moderation bots to automatically ban users who post unwelcomed content in their channels [42, 49].

While newly introduced automated content moderation tools like Crossmod [8] are useful for typical moderation tasks, they might not be suitable for community-specific tasks where micro norms should be enforced. This is because it is hard to find enough datasets to train the machine learning model that would perform the desired community-specific tasks. For example, when a subreddit wants to prohibit their members from writing "Merely indicating agreement conversation" comments (r/BlackPeopleTwitter) [9], it may be hard to train a machine learning model to learn this specific task due to lack of existing labeled datasets. In this work we present ModSandbox, a solution that utilizes automated moderation rules to reduce moderators' time and effort to review posts to resolve false positives and false negatives manually. Our approach helps refine automated rules to tune them as desired for their community. We believe our work could help community-driven sites' moderators better use automated moderation tools, especially for community-specific rules.

2.3 Importance of Human-in-the-loop Content Moderation in Online Communities

While various automated moderation bots are introduced to help reduce the workload of human moderators, humans are still needed in the loop of content moderation procedures. This is because humans are more skilled at fully grasping the context and intent of user-created content and better understanding the conceptual vision of a community [34]. Human moderators can also exchange authentic conversations with community members to resolve any misunderstanding on community rules or to revert moderation mistakes. Researchers also argue that because of the inherent complexity and ambiguity of moderation tasks, automated moderation is not sufficient in many cases [37]. This is especially true for user-driven moderation in small communities where moderators want to apply their own norms and policies.

Recent studies have explored different approaches in building effective human-in-the-loop systems for online content moderation. Vaidya et al. [46] found how providing visual analytics support can help human-in-the-loop content moderation by enabling transparent and communicative moderation practices. PolicyKit [50] enabled community users to write a small amount of code to automatically carry out policies on behalf of the community, which could be easily updated by the users' needs and intentions. A more simple form of human-machine collaborations is also introduced. For example, community members make use of third-party tools that visualize members' comments to help poll their opinion for community-level decision-making [11]. Other tools help online community members deliberate and reflect their opinions so that diverse perspectives can be considered in decision-making [29, 30].

While many automated moderation tools are introduced to help human moderators to automate part of their tasks, one of the main challenges for human moderators still remains: Many times these automated tools create new tasks for moderators, putting an additional burden on them [18]. To help reduce these burden in using human-in-the-loop content moderation tools, we investigated through surveys and interviews why and how using automated moderation tools is difficult for moderators, and identified design goals to reduce the human effort put into monitoring their communities and debugging and fixing automated tools.

2.4 System to Help Build Algorithmic Models for Classification Tasks

Configuring an automated moderation tool is a process for moderators to build algorithmic models to classify specific predominantly text content as they intended. Moderators are usually domain experts who understand how to maintain the community better than anyone else, but most of them do not have technical expertise in building models for text classification. In this section, we review previous research on systems that support users to customize their own classification models for their individual and specific goals, which has similar motivation to our work but in different contexts and applications.

There have been attempts to create systems to help people easily set up classifiers by utilizing a confusion matrix. EnsembleMatrix [45] supported end-users to create an ensemble model combined with multiple classifiers. It visualizes the classification results of each component classifier and ensemble classifier using heat-map confusion matrices. It allows users to track the change of the result of the ensemble model while they change the weights of each classifier with sliders. ManiMatrix [24] showed the performance of a classifier on a confusion matrix and allowed users to control its behavior by directly updating the parameter on the matrix, not the model parameter itself. When the user change a parameter on a cell, it highlights the changes of other parameters by color feedback. These systems help users understand how changes in parameters affect the results using immediate visual feedback.

Specific to text classification, some systems tried to help users extract meaningful words that represent a target class as features. These features can be used to train any algorithmic model to classify the documents. DUALIST [44] provided lists of words and documents that users can select to train a model. It showed the meaningful set of queries related to the target class first using the trained model to help users reduce cognitive loads for selecting features. Conversely, FeatureInsight [3] showed errors to help users to select features for the target class. It provides visual summaries that show the terms used in misclassified documents and bars next to each term indicating its number in the right and wrong documents. Also, the users can select a term and see the related words to expand their feature sets. These systems allowed users to review and choose the meaningful features based on their domain knowledge using their sorting algorithm. Our study designed and developed a novel approach using a sandbox environment to help users easily make a model for their text classification tasks, but specifically focusing on configuring an automated moderation bot for voluntary moderators in online communities.

3 SURVEY AND INTERVIEWS: KEY OBSERVATIONS

To better understand the current challenges moderators face when configuring automated tools for moderation, we conducted two web-based surveys and one remote interview study with Reddit moderators. We chose Reddit as a case study because it is one of the most popular sites on the web with over 130K active sub-communities, called subreddits, where each subreddit has its own volunteer moderators. We designed the first survey to ask about failure and frustration that are caused by the unexpected results of using AutoModerator, e.g., a shift in post topics due to COVID-19 pandemic, rapid growth in the number of active members, Reddit's policy update, a shift in members' stance or opinion about something, etc. The questions were targeted to find answers to questions that the research team could not answer through a literature review. The second survey was added to ask about specific experiences on false positives in keyword-based filtering caused by using AutoModerator based on the results from the first survey.

To recruit respondents for our surveys, we sent private messages with the survey link to moderators randomly sampled from a list of subreddits [1] via Reddit's internal mailing system. Fifty-two moderators participated in the first survey, and 50 moderators participated in the second survey. To analyze the survey results, we computed the number of answers for multiple-choice questions and "yes or no" questions to compare the number between different answers. For the free-form answer questions, one of our authors conducted thematic analysis [2] by extracting keywords from each answer and two other authors reviewed the original code and themes to resolve any issues.

Next, we conducted semi-structured remote interviews with six Reddit moderators to better understand the AutoModerator configuration process and distill the design requirements for a system to support it. Five moderators were recruited by sending emails to moderators who left their contact information in the survey. One moderator—the inventor of AutoModerator—was introduced by one of the interviewees. Each interview session took 40 - 70 minutes via an online conference call, and each participant was paid a $30 Amazon gift card for their participation. To extract the patterns of the configuration process from the interview transcriptions, five authors participated in an iterative coding process through multiple pairing sessions [15]. The authors were randomly paired for each session and coded one interview transcription. Any disagreement was immediately resolved through deliberative discussion. After coding all six transcriptions, all five authors gathered for four consecutive 2-hour meetings to interpret and find patterns of the code. They discussed until they reached a consensus on the final codebook and derived themes of the process and the design requirements. Based on the themes, an affinity diagram on expert moderators' moderation process was made as in Figure 2. The design requirements were used to identify the extra burdens in the process.

Below we summarize the findings from moderators' responses from both the surveys and interview. We assigned different symbols for the surveys responses (R1, R2) and the interview responses(R3), and numbered the quotes in the order of participation.

3.1 Unexpected False Positive Results Caused by AutoModerator are Difficult to Detect and Analyze

In the first survey, 45% of the respondents reported that they have experienced unexpected results or side effects after changing AutoModerator rules. Nineteen respondents in Survey 1 described the circumstances of the unexpected results, and 10 of them said they were caused by AutoModerator filtering more posts than they intended For example, respondent *"Got a rule slightly wrong leading to all posts being removed (R1-30)"*, and *"We attempted to filter out some common slur/unproductive words (cancer and shill, respectively) and noticed that a lot of good posts were getting caught up in the less offensive uses of those terms. (R1-37)"*. Two of the respondents in Survey 1 responded that the posts that

[1] https://www.reddit.com/r/ListOfSubreddits/wiki/listofsubreddits/

should remain were filtered instead of those that should be removed: *"Some of my rules backfired and had the opposite effect (R1-6)"*. These problems arise due to false positives where unintended posts are being filtered by AutoModerator. Other responses pointed out a syntax error or regular expression error as reasons for unexpected results.

3.2 Moderators Feel Frustrated Because They Cannot Debug AutoModerator Before Deploying

In the first survey, about 48% of respondents reported that they felt frustrated when configuring AutoModerator because no mistake is allowed after deployment. Five of them said they felt it most of the time. Eleven respondents in Survey 1 reported that they created a private subreddit to test basic configurations before deploying to avoid making mistakes. However, these private subreddits are not helpful when checking false positives because they do not have actual posts or comments created by the users. It is implied from responses that the private subreddits are only used to check style changes or basic syntax errors. The rest of the respondents in Survey 1 reported they did not test AutoModerator configuration before deployment or test in production. These results motivated us to build a virtual sandbox environment to test automated moderation bots, which can help resolve the moderators' frustration.

3.3 Moderators Only Address False Positives After They Happen

Informed by the first survey and with the sandbox idea in mind, we narrowed down the problem space to false positives in keyword-based filtering in the second survey. We chose to focus on moderation by keyword filtering, where moderators set up rules that filter posts with particular keywords or a combination of keywords because it is known to be one of the most common usages of automated moderation bots [18] (92% of respondents in Survey 2 reported that they perform keyword filtering with AutoModerator). We asked Survey 2 respondents to share detailed descriptions of their experiences in keyword filtering. Examples of false positives with keyword filtering included *"a url had kys in it (short for kill yourself) (R2-5)"* and *"Overlap between reclaimed slurs and trolls using slurs, mainly (R2-15)"*.

Eighteen respondents in Survey 2 shared how they tackle false positive problems. Fourteen of them mentioned that they manually inspect AutoModerator-filtered contents one by one and determine whether to revert the removal. Two of them mentioned that subreddit users report false positives to moderators, and they revert them from being filtered. These patterns were observed in our interview study as well. Overall, moderators dealt with false positives by putting in manual labor or relying on user reports, which vitiates the meaning of automating moderation tasks. Moreover, these workarounds are post hoc treatments which happen hours or days *after* the false positives affect their communities.

3.4 We Found Four-Step Process of AutoModerator Deployment and Its Challenges

We asked the interviewees to elaborate on how they create and update their AutoModerator configuration and any heuristics they use to alleviate any issues that arise. We found that moderators follow four main steps when configuring AutoModerator: 1) be motivated to create an automated rule, 2) examine and discuss candidate rules, 3) configure and deploy a rule, 4) evaluate the rule by monitoring the community or receiving user feedback. We found that there could be multiple cycles inside the model from Step 2 to Step 4 because the rules are often updated if there is any issue detected in Step 4. Figure 2 shows the four steps that we derived from the interview.

Step 1. All participants said that they are motivated to create a new AutoModerator rule when they need to perform repetitive tasks. Creating a new AutoModerator rule consist of recognizing posts or authors that violate the community rules and taking action such as removing the post or leaving a comment. Sometimes they find some patterns from the contents and their metadata. R3-3 said *"in some of the subreddits, we noticed that posts that are very short tend to be ones that we take down because they're not very good. And so we set a thing that if it's a short post, then it automatically takes it*

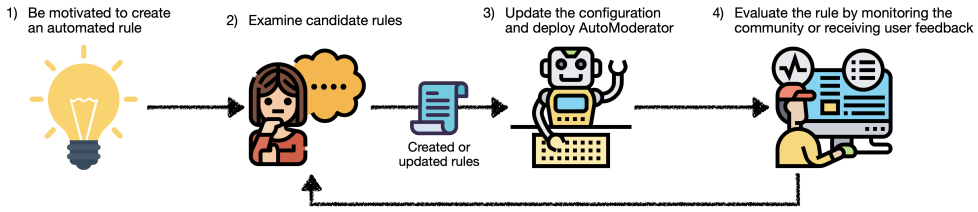

Fig. 2. Expert moderators' four main steps for configuring and updating Reddit's AutoModerator.

down." R3-2 mentioned *"Like we have a lot of questions about people buying violins on Amazon, which is usually not a good idea... I kind of at this point, very simply have automoderator look for the word Amazon, because it's kind of a thing that doesn't really come up otherwise."*

Step 2. After deciding to automate a task, moderators either test an AutoModerator configuration in production and then quickly remove it (so that it has no significant effect on the subreddit) or discuss the configuration with other moderators to collectively gauge if the rule would act as expected. During this process, they pick up common patterns from target posts or think of patterns based on their heuristics to use in writing AutoModerator rules. R3-2 mentioned *"I think we just started in the beginning with whatever, just what we could think of."*. They try to find obvious patterns that only the target posts have and expand the patterns to catch as many targets as possible. This was done internally using their memory and heuristics.

Step 3. Next, they update the AutoModerator configuration by applying the patterns they decided to capture with the bot and deploy it live in their community. Interviewees reported different update cycles depending on the situation of their community. For example, R3-3 said *"In the heyday of, of one we were dealing with these bots. I mean, I was adding stuff to the list, you know, every day or two every couple of days... it really petered off when the subreddit got quarantined"*.

Step 4. Lastly, they keep monitoring the community for a few days or weeks to see if the new rule works as expected. R3-2 said *"For the most part, since the initial setting up, it's just kind of looking at what doesn't get caught."* There are two issues moderators are mainly interested in resolving: AutoModerator misses the desired posts (false negatives) or it filters the wrong ones (false positives). To discover false negatives, they pay attention to almost every new post or user report, which is a similar process to creating a new rule. On the other hand, false positives are detected via feedback from community users. They normally get mod mail when users report that their posts have been removed by AutoModerator and check AutoModerator logs or a spam queue, where removed posts are recorded.

They keep iterating this four-step process to evaluate and update the rule if necessary. This iteration helps moderators reduce both false positives and false negatives through fine-tuning their AutoModerator rules. Additionally, we found that moderators have difficulty in managing the complex rules and reasoning why a post is filtered or not. The more rules in the configuration, the harder it becomes to understand how many posts are filtered by each rule. R3-4 said *"sometimes it can get really messy, and you know, you don't really know anymore, what it's doing, or there's multiple rules for the same thing."* Even when they find a rule that filters out a post, they occasionally failed to understand why, especially if the rule contains many keywords or regular expressions. R3-4 stated *"there's a few times that it's picked out comments that I can't figure out what's the word. Every so often."*

4 MODSANDBOX: A RAPID TESTING SYSTEM FOR AUTOMATED MODERATION RULES

This section describes four design goals for a sandbox system to help moderators test automated moderation bots, guided by findings from the surveys and interviews. We then introduce ModSandbox (Figure 3), a system built to support online community moderators with rapid testing and validation of their automated rules.

4.1 Design Goals

Based on the key observations from our surveys and interviews, we present two main goals for building ModSandbox, our rapid testing system for automated moderation rules.

- Reduce effort and time in detecting false positives and false negatives.
- Reduce effort and time in debugging and fixing automated rules before deployment.

Below we present more specific design goals that have guided the design of our system.

DG1. *Provide a sandbox to enable prompt evaluation of a configuration without affecting the posts and comments in real communities.* According to our surveys, moderators are frustrated because false positives and false negatives are hard to detect, and debugging configurations of the automated rules to fix them is even harder. Hence, we propose a sandbox environment as a solution, which imports real posts from the moderator's community and helps moderators evaluate the configurations of automated rules in a simulated environment without affecting posts and comments in their real community.

We note that the sandbox environment uses existing posts to help find false positives and false negatives. That is, the effect of automated rules in future posts is not directly assessed. Therefore, these sandbox environments are particularly useful when a community's members' characteristics and their posts are likely to remain similar over time, letting the configured automated rules that work well on the past posts work as well on the future posts.

DG2. *Enable users to have a space to collect posts instead of having to memorize all of them when trying to identify their patterns.* Moderators in our surveys and interviews had to monitor their communities frequently, manually tracking and memorizing the false positive posts to find patterns of them in their heads. However, this is a time-consuming and mentally demanding task. Providing a space for moderators to copy and save targeted posts for the later examination can reduce the cognitive load in tracking them. Once they collect enough posts, moderators can look at the collection of posts at once to find patterns by comparing them to each other.

DG3. *Provide methods to quickly discover false positives and false negatives.* Moderators in our surveys and interviews found it hard to detect false positives because they happen only once in a while, easily being buried in other posts. This is true for false negatives as well because malicious users can circumvent banned keywords to violate community rules [18]. Moderators may only find these false negative posts that are not filtered by automated rules after the posts are exposed to other members, making them uncomfortable. Moderators are motivated to find these false positives and negatives as soon as possible because even though they may happen once in a while, they easily affect the community in a negative way, sometimes to a serious degree. As a workaround, moderators constantly monitor their community or wait for user reports to detect false positives and false negatives, which is time-consuming and mentally demanding.

Therefore, we propose to help moderators easily find false positives and false negatives through the support of natural language processing (NLP) techniques. We suggest using NLP techniques because they are good at categorizing

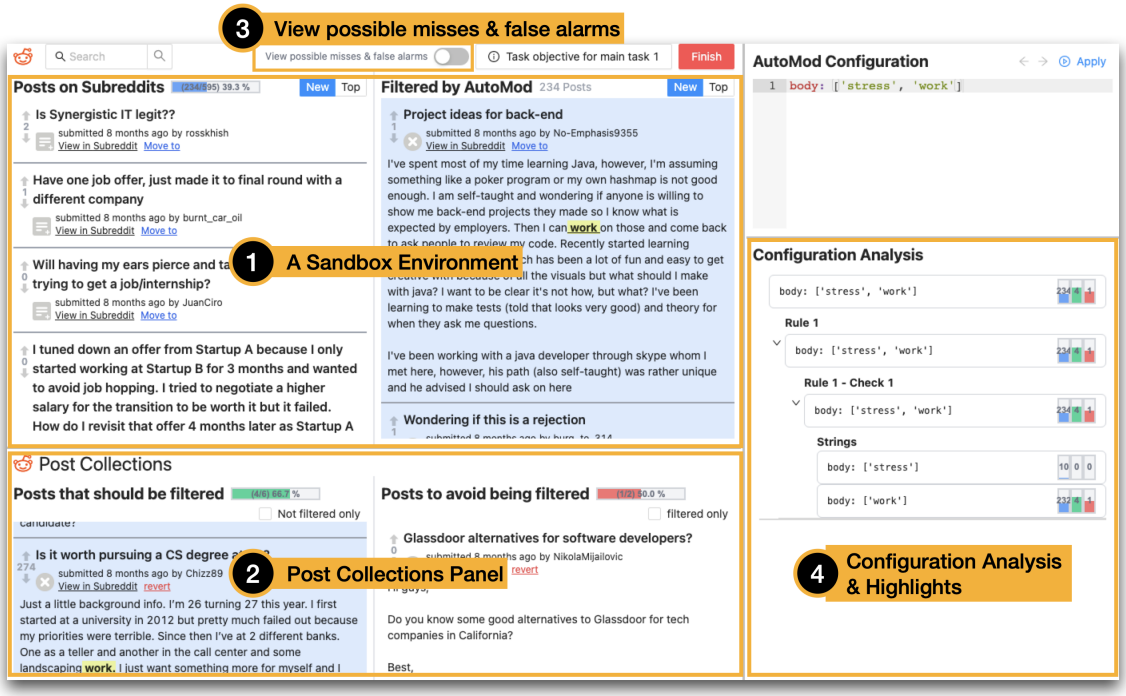

Fig. 3. An Overview of the four main features of ModSandbox. ❶ is a "Sandbox Environment" where a moderator can import all the posts from their community. ❷ is the "Post Collection" area that helps moderators to save targeted posts for later inspection instead of having to memorize their content. ❸ is a toggle button that rearranges the posts in the sandbox area from the most "Possible misses and false alarms" to the least. It helps moderators to more quickly find possible misses (false negative) and false alarms (false positives). ❹ is the "Configuration Analysis" panel that helps analyze how the rule affected the posts in the sandbox. It is linked to the sandbox area to highlight keywords for micro-level support of debugging each configuration.

a large number of corpora in real-time. Since the number of posts in the sandbox could likely exceed hundreds and thousands, NLP techniques become useful to category them into possible false negatives and possible false positives to help moderators discover them quickly. By assisting moderators in discovering false positives and false negatives efficiently, it can result in reduced time and effort to constantly monitor their communities.

DG4. *Enable analyzing the impact of an automated rule at both micro and macro level through simple methods.* According to our interview results, knowing how many posts are affected by a configuration can help assess whether an automated rule is functioning as intended. Therefore, we suggest that macro-level supports to analyze the impact of configurations can satisfy moderators' needs. For example, this could be done by simply visualizing the number of posts being affected by the configuration compared to the total number of posts. We also found from the interviews that often it is hard to tell the reason why a post was filtered by an automated rule. For example, among many banned keywords like swear words, it may be hard to tell which word triggered a post to be automatically removed. Therefore, we propose micro-level supports to analyze which part of a post caused the automated rule to filter it. This functionality may help even the moderators without a programming background to easily debug and fix their automated rules.

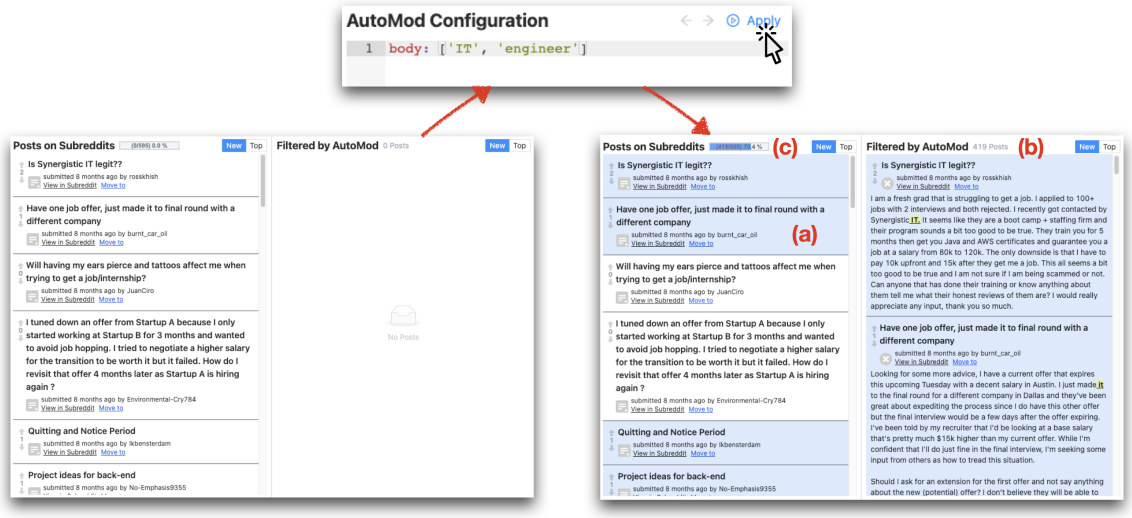

Fig. 4. Shows how to use the Sandbox environment. The left figure shows the sandbox right after importing posts from a community. When a user clicks on the "Apply" button after writing their rules in the "AutoModerator Configuration" panel, (a) the background turns blue for the posts filtered by the rules, (b) "Filtered by AutoMod" gathers them in a separate panel for easy browsing, and (c) a ratio bar shows how many posts are filtered by the rule.

In the following subsections, we describe the four main features of our ModSandbox, a rapid testing system for automated moderation rules, that are guided by the aforementioned design goals.

4.2 Feature 1: A Sandbox Environment

ModSandbox provides moderators with an isolated sandbox environment (❶ in Figure 3), which allows moderators to virtually test their automated rules on posts that already exist in their communities. This helps them identify or predict any issues with the rules without affecting posts in their actual communities. Figure 4 shows an example of using our sandbox environment. A moderator imports posts from a subreddit named r/cscareerquestions to the panel "Posts on Subreddits". Then they write an automated rule in the "AutoMod Configuration" panel to filter any post with the words 'IT' and 'engineer'. After they click on the "Apply" button, every post that includes the keywords turns blue to provide a visual comparison between filtered and non-filtered posts. Moderators can also see the filtered posts in "Filtered by AutoMod.", which gathers them in one place for easy browsing. This rearrangement and coloring of posts help moderators understand which types of posts are being affected by the rule. Additionally, a horizontal bar right next to the word "Posts on Subreddits" presents the ratio of filtered posts. In the figure, we can observe that over 70% of the posts include two keywords. Removing posts with these keywords may be a bad idea because it removes most of the posts in the community. That is, this ratio bar helps moderators understand the macro-level effect of the automated rules so that they can assess whether the rules are functioning as intended or harming the community.

4.3 Feature 2: Post Collections Panel

The "Post Collections" panel (❷ in Figure 3) facilitates the evaluation of an automated rule. Moderators can collect posts that are useful to evaluate their rules such as *posts that should be filtered* or *posts to avoid being filtered* in separated

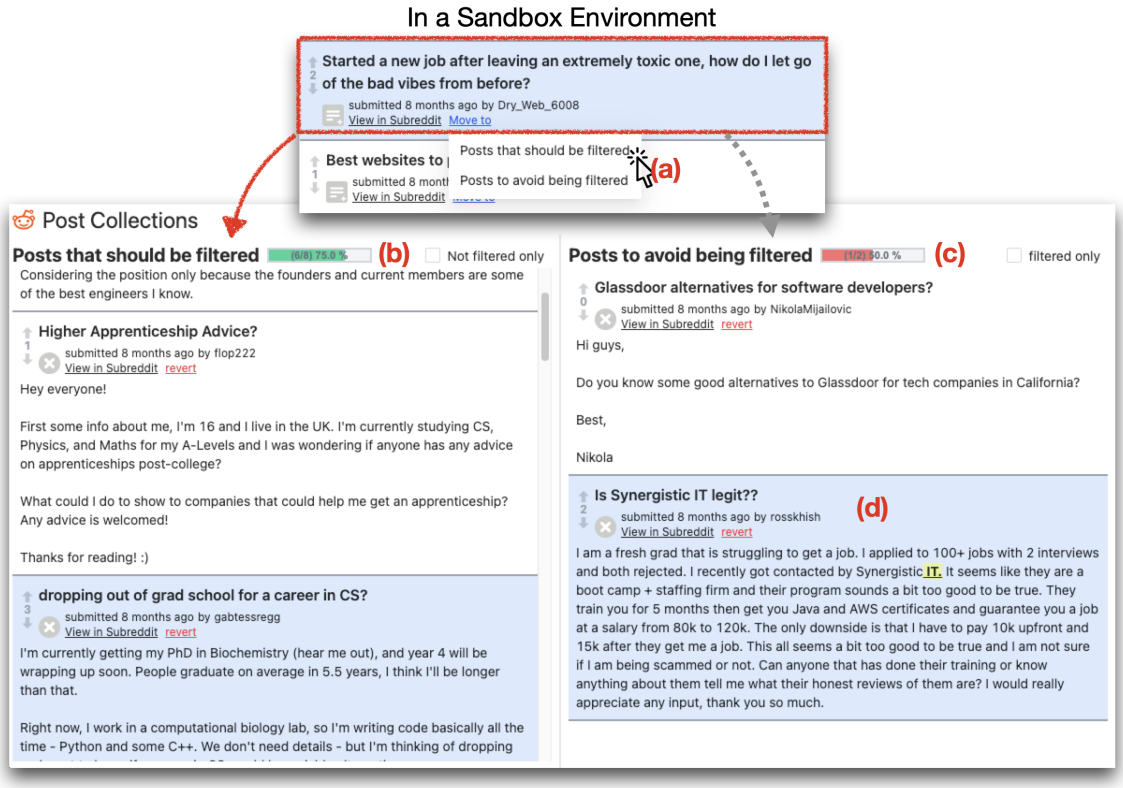

Fig. 5. Post Collections panel. (a) The users can move posts from the Sandbox Environment to one of the Post Collections panels: "Posts that should be filtered (red solid arrow)" and "Posts to avoid being filtered (gray dashed arrow)". (b, c) The green and red bars show the ratio of the filtered ones. (d) The filtered posts by the automated rules are marked blue in the Post Collections panel.

spaces. Figure 5 shows an example of using the Post Collection panel. A moderator can move posts that they want to filter with an automated rule to the "Post that should be filtered" panel. Once enough posts are collected, the moderator can use this panel in two ways. First, they can browse through the posts to find patterns that could be useful to write an automated rule, e.g., find common keywords among the posts collected. Second, they can see how many collected posts are actually being filtered with the current automated rule ((b) in Figure 5). If the number of posts being filtered is too low, they may want to update the automated rule to filter more posts.

A similar practice could apply to using the "Post to avoid being filtered" panel. A moderator can collect posts that should not be filtered in this panel to find common patterns among them. Then they can write an automated rule that would avoid filtering these posts. The moderator can monitor the red bar ((c) in Figure 5) in this panel to see if the current rule is successfully avoiding filtering posts in this panel. For example, in the figure, since 50% of the posts in this panel are being filtered, the moderator might want to update their automated rule to reduce this number.

This Post Collection panel is also used for the next feature: View Possible Misses and False Alarms.

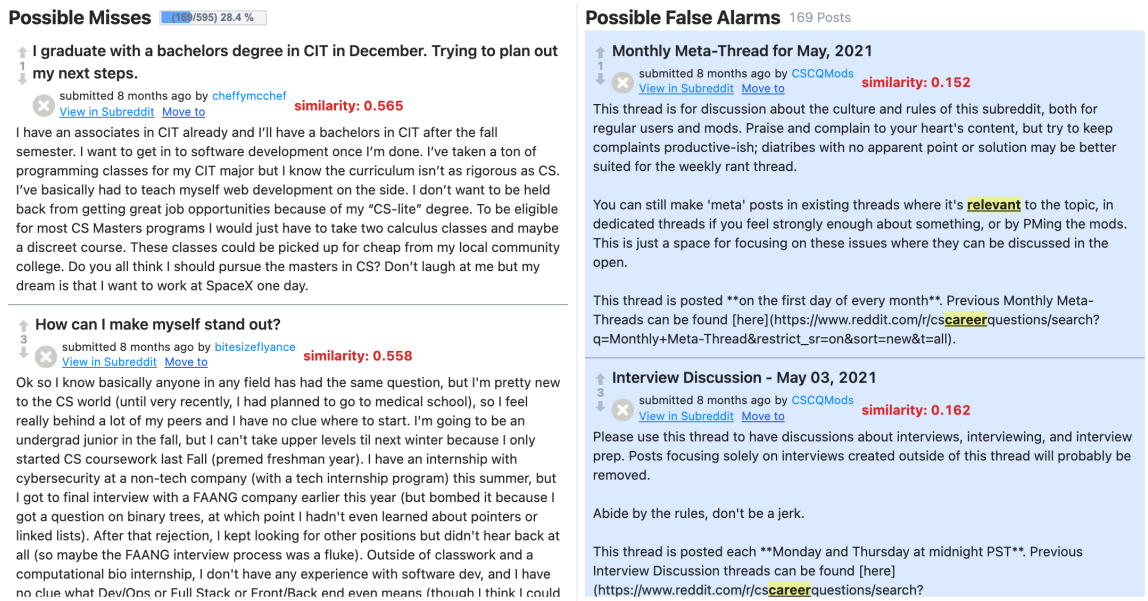

Fig. 6. Example of possible misses (false negatives) and false alarms (false positives) of the configured rules in Task 2 of our main user study. Participants were guided to detect posts about asking whether or how to get CS-relevant jobs without CS-relevant degrees. The more probable posts that are being missed are listed at the top (e.g., similarity 0.565 is larger than 0.558), and the opposite happens for the false alarms (similarity 0.152 is smaller than 0.162). The similarity values are hidden in the actual interface.

4.4 Feature 3: View Possible Misses and False Alarms

ModSandbox provides the "View Possible misses and false alarms" feature to help moderators quickly find issues with their automated rules, i.e., false positives and false negatives. When moderators activate this feature by toggling a button (❸ in Figure 3), possible false positives (equal to false alarms) and false negatives (equal to misses) are presented from the most probable to the least. This feature helps moderators to quickly find possible false positives or false negatives without having to browse all the posts in the sandbox.

To sort all posts from the most probable miss to the least and the most probable false alarm to the least, we use the Universal Sentence Encoder [6], a set of NLP models that encode sentences into embedding vectors. More specifically, all the posts in the sandbox are encoded into vectors, and then their distance is compared with the posts in "Posts that should be filtered".

If a filtered post is semantically far different from the posts in the "Posts that should be filtered", it is likely that the post is a false alarm. Motivated by this intuition, we treat posts that are filtered but are different (by vector distance calculation) from the posts in "Posts that should be filtered", as *possible false positives*. Conversely, we treated posts that are not filtered but are similar (by vector distance calculation) to the posts in "Posts that should be filtered", as *possible false negatives*. For vector distance calculation, an average of the vectors of the posts in "Posts that should be filtered" are used as a reference point. This happens at the time of import to reduce delay during the test time, and the reference is recalculated whenever moderators add a new post to the "Posts that should be filtered" panel. For example, as shown in Figure 6, a non-filtered post with the closest distance from the reference point comes at the top of the "Possible Misses" panel. The farthest non-filtered post comes at the bottom of this panel. This way, the more important posts

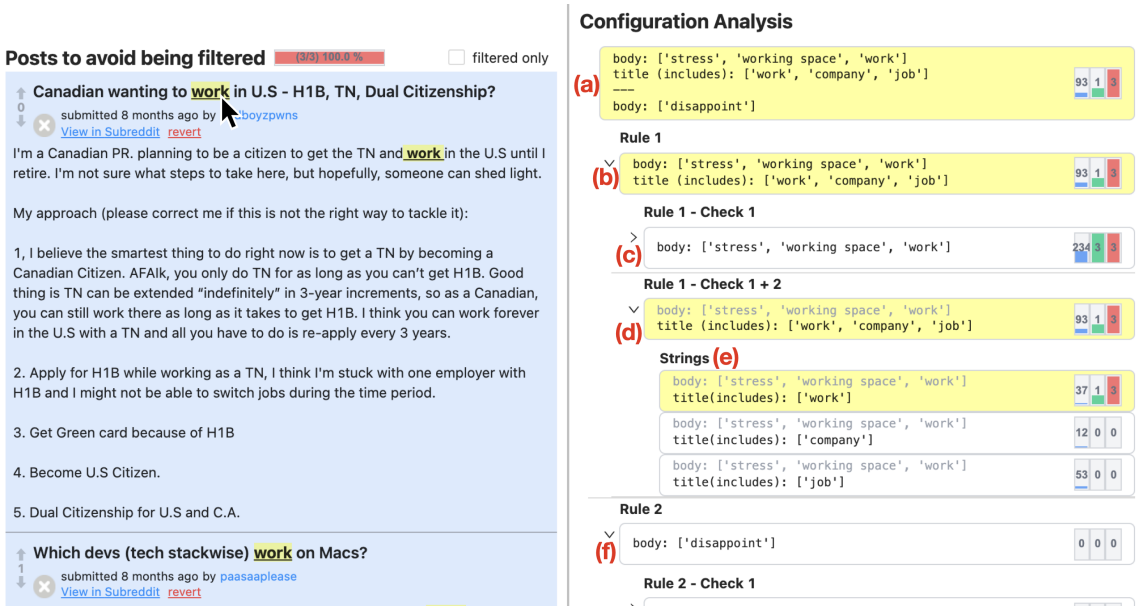

Fig. 7. An example of the Configuration Analysis and the Highlights feature. Each labeled box with rounded corners on the right side represents a part of a configurated rule. The three embedded vertical bar graphs on the right side of each rounded box show the number and ratio of filtered posts in three different types of posts: "Posts on Subreddits", "Posts that should be filtered", and "Posts to avoid being filtered", respectively.(a) shows an AutoModerator configuration that consists of multiple rules: (b) and (f). Rule (b) detects intersection of posts detected by checks (c) and (d). Check (c) finds the posts that have any of 'stress', 'working space', and 'work' in the body. Among the posts detected by check (c), check (d) detects the posts that includes any of 'work', 'company', and 'job' on the title. (e) shows the impact of individual strings in the check (d). The Highlight feature emphasizes specific part of posts affected by the configuration. As shown in the left side of the figure, when a user hovers the cursor on the word "work" in the post title, relevant items (the rounded boxes) are highlighted on Configuration Analysis panel.

are seen first to the moderators, helping them to find clues to update their automated rules to filter these misses. We excluded some metadata like the account age, which Reddit moderators utilize for moderation so that we can simplify the interface to focus on a single user study task of keyword filtering.

4.5 Feature 4: Configuration Analysis and Highlights

ModSandbox helps moderators analyze the impact of their complex rules through "Configuration Analysis" and "Highlights" features (❹ in Figure 3). First, Configuration Analysis represents rules in a tree-structured format, allowing moderators to easily analyze them one by one. As shown in the right side panel in Figure 7, AutoModerator supports multiple rules that have multiple checks. A single check corresponds to an attribute of the posts such as body or title and filter them with a single list of strings to filter. Using the Configuration Analysis feature, moderators are able to evaluate the impact of each rule, switching through check or string. The three bar charts on the right indicate how each part of rules impacts posts in "Posts on Subreddits", "Posts that should be filtered", and "Posts to avoid being filtered", respectively. The bar charts are used to assess the impact of each check and string by understanding how they affect the posts. For example, looking at the three bar charts in Figure 7(d), more posts in the "Posts to avoid being filtered (red)" are being filtered compared to the posts in "Posts that should be filtered (green)". Because it seems adding the

second check "`title (includes): ['work', 'company', 'job']`" caused this adverse effect, and a moderator may decide to get rid of this second check to update the automated rule.

"Highlights" presents a quick visualization of which part of the post is affected by the current rule. For example, if the rule filters out posts with the word "relevant" in the body, the word "relevant" in the post is highlighted (see Figure 5). This feature helps moderators to quickly and easily debug their automated rules.

5 EVALUATION AND RESULTS

To verify whether ModSandbox can support moderators with rapidly testing automated moderation rules and to evaluate the usefulness of the features we implemented, we conducted user studies with 10 active online community moderators. We first built a baseline system that simulates a general process of creating Reddit's AutoModerator rules. Then, we built ModSandbox by adding the proposed features in Section 4 to the baseline system. Both the baseline and ModSandbox allow the users to do typical things they would do when moderating their subreddits: browsing community posts, searching posts by words or phrases, and sorting posts by newest and highest votes.

To select the subreddit for the user study, our criteria included (1) whether the community is active, (2) whether posts are mostly text-based as our scope focuses on keyword-based moderation, and (3) whether we can understand the community norm easily so as to make plausible hypothetical moderation tasks for the user studies. We crawled posts from r/cscareerquestions, a subreddit where members post questions about computer science careers from May 1st, 2021 to May 7th, 2021, and imported them in both ModSandbox and the baseline.

5.1 Participants

We recruited 10 participants (Table 1), seven Reddit moderators (five males and two females) from the U.S. and three non-Reddit moderators (three females) in charge of Korean online communities. All were familiar with reading and writing English, so they participated in the same English-based user study as others. We sent a recruitment advertisement to Reddit moderators through the mod mail, which is a message system within the Reddit platform. We contacted moderators of subreddits randomly sampled from a list of subreddits in r/ListOfSubreddits.[2] The non-Reddit moderators in South Korea were recruited via word-of-mouth. We expected the non-Reddit moderators to represent voluntary moderators outside Reddit. While they might not be familiar with Reddit, we confirmed their moderation practices and challenges aligned with those on Reddit, while they might have unique moderation experiences.

Additionally, we ensured that we have moderators both with and without experience using AutoModerator. Five of the participants (P1, P5, P7, P8, P10) had experience configuring the AutoModerator while the others (P2, P3, P4, P6, P9), including Korean community moderators, had little or no AutoModerator experience. The recruitment method and the study design were approved by our institution's IRB policy.

5.2 Study Procedure

The study sessions were conducted remotely through Zoom, one of the top videoconferencing platform at the time of research. Before the session, participants filled out a consent form and a survey about their background including AutoModerator experience (asked only to Reddit moderators) and familiarity with programming (related to how fluent they might be in understanding the authoring rules and their syntax). Each study lasted about two hours, and each participant received a $30 Amazon gift card per hour as compensation.

[2] https://www.reddit.com/r/ListOfSubreddits/wiki/listofsubreddits/

	Age	Gender	Moderator periods	Prior experience			Familiarity		Condition for User Study
				Reddit Moderator	AutoModerator	Programming	Subreddit	Language	
P1	35-44	M	over 5 years	Reddit	occasionally	basic concepts	2	2	Experienced
P2	18-24	F	6 months - 1 year	Non-Reddit	never	basic concepts	1	1	Novice
P3	25-34	F	under 6 months	Non-Reddit	never	No knowledge	1	1	Novice
P4	18-24	F	6 months - 1 year	Non-Reddit	never	frequently	1	1	Novice
P5	25-34	F	1 - 2 years	Reddit	occasionally	basic concepts	3	4	Experienced
P6	25-34	M	under 6 months	Reddit	done it once	No knowledge	1	2	Novice
P7	25-34	M	2 - 3 years	Reddit	occasionally	frequently	2	2	Experienced
P8	18-24	M	2 - 3 years	Reddit	occasionally	a few programs	4	3	Experienced
P9	45-54	M	6 months - 1 year	Reddit	done it once	a few programs	1	1	Novice
P10	18-24	M	1 - 2 years	Reddit	most of the time	a few programs	3	5	Experienced

Table 1. Background Information of Study Participants. Familiarity scores range from 1 (Not at all) to 7 (Very much). The subreddit we asked of familiarity is r/cscareerquestions and the language we asked of familiarity is YAML.

5.2.1 Tutorial on How to configure AutoModerator (20-30 minutes). Before entering the main task, we explained what is expected in the main tasks and taught or reviewed how to write Reddit's AutoModerator rules. Next, we gave a walk-through tutorial on how to use features in the baseline system and ModSandbox. The ModSandbox tutorial was given in between transition from the baseline system to ModSandbox to not overload the participants with too much information at once. The order of tutorials was determined according to the order of the main tasks. The rationale behind the order of the main tasks is explained in the next subsection.

After the tutorial session, we asked the participants to solve quizzes about the study to make sure they understand the purpose of the study, how to configure AutoModerator, and how to use the systems. If they got wrong answers, we helped them find the right answer and checked if they understood everything correctly. This step ensured that everyone was equipped with adequate rule-authoring skills to perform the main tasks. We also provided the participants with two reference documentations: one with descriptions of ModSandbox features and the other explaining the AutoModerator configuration grammar, which they can freely access during the main tasks. The tutorial and documents successfully helped most participants to write an AutoModerator configuration themselves, except P3 who had difficulty writing the rule due to lack of programming knowledge.

5.2.2 Main Tasks (60-80 minutes). The posts from r/cscareerquestions were imported to both the baseline system and ModSandbox in advance to reduce the total study time. Note that the posts imported are already moderated with the community's actual rules, e.g., slurs and swear words are already filtered out. For the main study, each participant was given two different tasks on writing AutoModerator rules for realistic but hypothetical moderation scenarios. More specifically, we created hypothetical moderation objectives instead of utilizing the subreddit's actual rules because of two major reasons. First, it leaves plenty of posts that could be moderated with our hypothetical rules, because we chose the hypothetical rules that do not overlap with the community's real rules. As the imported posts have already gone through moderation, asking participants to apply the same moderation rules would leave few posts to be actually moderated. Second, by using posts that are already moderated with the community's real rules, the participants do not have to be exposed to mentally abusive posts including slurs or swear words.

The two main tasks we set were as follows:

- Task A: Write an AutoModerator rule to detect posts about asking whether or how to get CS-relevant jobs without CS-relevant degrees.
- Task B: Write an AutoModerator rule to detect posts that include keywords related to COVID-19.

For each task, we provided a background scenario and three example posts that meet the moderation objectives to help participants engage with the tasks. We introduced those three example posts as posts that had already been manually filtered by other peer moderators. The main tasks for the participants were to write AutoModerator rules that would filter similar posts with the example posts. Providing example posts helped the participants to have similar internal criteria on how they understand the hypothetical moderation scenarios.

The study was carried out as a within-subjects study, where each participant used both the baseline system and ModSandbox. We provided Task A and B in a randomized order for each participant but provided baseline and ModSandbox in a fixed order. That is, for each moderation task, participants first started with the baseline system where they were asked to create AutoModerator rules as accurately as possible, and then they move on to using ModSandbox to update the rules they have written so far. In each study session, we emphasized that the rules written in the baseline system should be in the form of their best attempt before they move on to using ModSandbox. Note that we did not randomize the order of the systems because we tried to observe how the moderators create their automated rules with the baseline, which simulates the current practice, and how they can further modify and improve their rules with ModSandbox. We simulated the current practice first because using ModSandbox first would provide too many hints on how to update rules for the baseline system, e.g., the participants are already exposed to the probable false positive and false negative posts. Since we did not randomize the order of the systems, we did not statistically compare performance differences between the two systems. We rather focused on interpreting the usage patterns of the built-in features of ModSandbox and quantifying how much the participants could improve an already written AutoModerator rule by using ModSandbox. The usefulness of ModSandbox's features was observed by the researchers through post surveys and interviews.

5.2.3 Post Surveys and Interviews (10-20 minutes). After the main tasks, participants took part in a survey about their main tasks experience; they answered 7-point scale questions on how useful ModSandbox's features were in each main task as well as the overall usefulness of ModSandbox. Once they submitted the survey, we verbally asked them about their strategies using ModSandbox and how they would use ModSandbox in actual moderation tasks for their communities. We also asked them for feedback on how the system could be improved.

5.3 Evaluation Method

We analyzed the study results based on various observations on the moderation processes using both the baseline system and ModSandbox. We examined the system logs and survey/interview responses, which explained the differences between the moderation processes when using the two systems. We intentionally chose not to quantitatively compare participants' rule accuracy between the system conditions, because each participant had different internal criteria about the given moderation tasks, resulting in no ground truth to exist for false positives or false negatives. In fact, the interpretation of moderation rules is inherently subjective, and the representation of these subjective rules into computationally automated rules will vary as well. We instead focus on observing how moderators implement their own moderation strategies and criteria using our system in a case study manner.

To verify if we can create ground truths for false positives and false negatives in the two main tasks, we hired two external annotators to label the test dataset with the given moderation scenarios: filtering posts about CS-relevant jobs (Task A) and filtering posts about COVID-19 (Task B). The posts to be filtered were labeled as 1 and the posts not to be filtered were labeled as 0. As a result, the inter-rater reliability measured with Cohen's Kappa was 0.45 for Task A and 0.67 for Task B. The scores were low even after having an asynchronous session (via email) to reach an agreement.

This was because each annotator had different internal criteria for each scenario. For example, Annotator 2 considered any post mentioning the usefulness of enrolling in a "bootcamp" to be filtered in Task A, while Annotator 1 did not agree with it. While Task B was slightly more objective, the annotators still had disagreements between their labels. For example, Annotator 1 considered any post mentioning "lockdown" to be filtered in Task B, while Annotator 2 did not agree with it. Hence, we concluded that forcing convergence to a single ground truth is not reasonable for both scenarios.

Instead of comparing the number of false positives and false negatives to a ground truth, we observe other useful quantitative measures as follows: the change in the number of filtered posts while using both baseline system and ModSandbox, the change in the number of used keywords using both baseline system and ModSandbox, the average complexity of the final rules using both baseline system and ModSandbox, and the average usefulness scores of each feature in ModSandbox. We also report participants' survey and interview results that help understand the observed differences using the baseline system and ModSandbox. The detailed results are discussed in the following subsection.

5.4 Results

In this section, we summarize our findings on participants' system usage and rule-making patterns employed with both the baseline and ModSandbox. We observed different patterns that highlight the difference between the two systems. Then we report the summarized results of post survey we asked after the user study.

5.4.1 ModSandbox Provides More Structured Rule Configuration Process through Rapid Evaluation and Update. From the user study, we found that participants create their own structured rule configuration process using ModSandbox, and iterates their process until they are mostly satisfied with the rule. We note that this iteration of updating rules takes days or weeks in a conventional moderation environment. It was possible to reduce this time significantly using ModSandbox because of two main reasons. First, past posts imported in the sandbox environment represented the future posts when assuming there is no large change in the community—not having to wait days or weeks to collect future posts. This is shown in Figure 10 where the rules configured looking at current posts apply well to future posts. Here, the current posts are the posts provided in the user study, and the future posts are posts generated a week after the current posts. Second, the features provided in ModSandbox improve the efficiency in analyzing and predicting the effects of automated rules, especially in terms of finding and resolving false positives and false negatives.

System usage pattern in Baseline. With the baseline system, we observed that participants came up with keywords by utilizing task descriptions, imaginations, and the example posts provided—utilizing some or all of them. Except for one participant who only used the keywords from the task description, nine participants expanded their keyword set outside the task description. Then, they guessed which and how many posts contain the keywords. Sometimes, the search feature was used to guess the impact of a keyword, and the participants decided whether they will include it as a filtering word. The search feature was used 3.58 times on average ($\sigma = 2.19$) among eight participants who used it.

Rule-making pattern in Baseline. To evaluate how complex each rule is, we computed the average complexity of the final rules by counting the number of rules, checks, and strings. By comparing the complexity of the final rules, we observed that the rules created in the baseline systems were relatively simple compared to ModSandbox (Table 2). Most participants made a single list of keywords and phrases rather than advanced combinations of units. In specific, five moderators from Task A and nine moderators from Task B submitted a single-rule and single-check configuration. For example, P5 submitted a rule that detects any of 'change', 'degree', 'machine learning' and 'worth' in the posts for Task

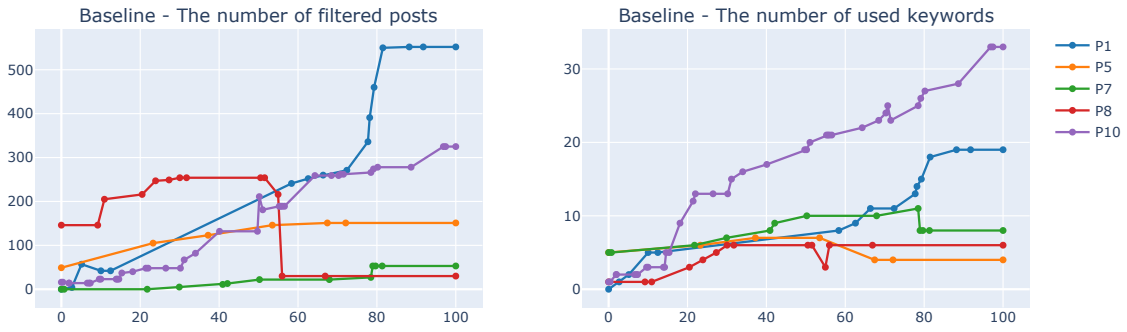

Fig. 8. For the baseline system, the number of posts being filtered by the experienced participants' configuration (left) and the number of keywords in the configuration (right) are shown in the time dimension. Because all participants had different ending times for their task, the total task times are normalized from 0 to 100 for better comparison. Both graphs show the temporal change where both the number of filter posts and the number of used keywords increase over time.

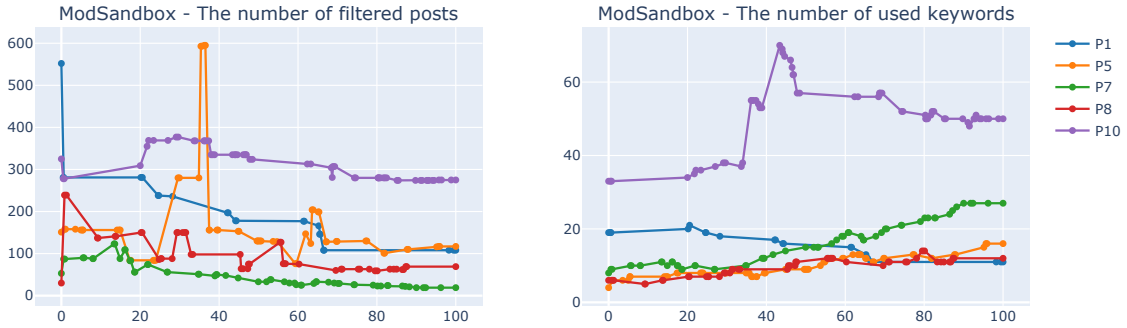

Fig. 9. For ModSandbox, the number of posts being filtered by the experienced participants' configuration (left) and the number of keywords in the configuration (right) are shown in the time dimension. Because all participants had different ending times for their task, the total task times are normalized from 0 to 100 for better comparison. Both graphs show the temporal change where the number of posts being filtered decreases. This implies that the participants found a way to reduce false positives using ModSandbox while they could not use the baseline system. Both graphs show more number of attempts to update rules compared to using the baseline system, thanks to the features in ModSandbox that help try writing different rules.

A. We also note one exceptional case where one moderator (P10) submitted a five-rule and nine-check configuration in Task A. An interesting pattern we found is the increasing number of filtered posts and filtering keywords over time from the changelogs of experienced participants (P1, P5, P7, P8, P10) (Figure 8). This complies with the findings from the surveys where moderators responded that they often create AutoModerator that filters more posts than expected due to not having any supportive tools to fine-tune it.

System usage pattern in ModSandbox. In ModSandbox, participants devised their own structured process to evaluate and update AutoModerator rules by creating their own combination of usage patterns of ModSandbox's features. All participants prefer turning on "View possible misses and false alarms" through the study session to see the posts that are likely to be false positives or false negatives. Their process to update rules mostly began with finding actual false positives and false negatives using the feature. A common structured process was as follows. Eight participants (P1,

	Task A		Task B	
Unit	Baseline	ModSandbox	Baseline	ModSandbox
Number of rules	1.4 (1.26)	**2 (1.86)**	1.4 (0.70)	**1.6 (0.70)**
Number of checks	2.2 (2.44)	**3.7 (3.47)**	1.8 (1.03)	**2.6 (1.96)**
Number of strings	10.6 (9.24)	**16.8 (14.02)**	5.9 (3.73)	**8.1 (3.75)**

Table 2. Average complexity (and standard deviation) of the final rules configured by the participants

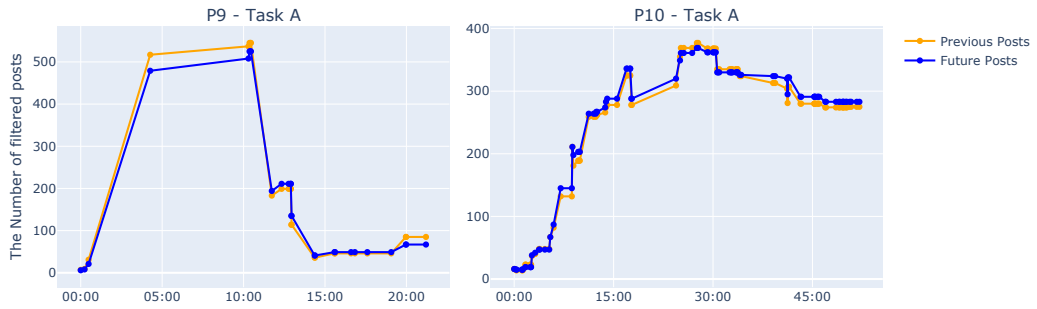

Fig. 10. The number of posts filtered by the configured AutoModerator rules of P9 and P10, who wrote the most simple rule and the most complex rule, respectively. The previous posts (orange lines) are the test posts given in the user study, and the future posts (blue lines) are the posts not given to the participants (posts created one week after the test posts). The orange lines aligning with blue lines indicate that the rules configured by the participants operate as intended for future posts even if they have not seen them.

P4, P5, P6, P7, P8, P9, P10) first reviewed possible misses and false alarms that our algorithm recommended and found actual false positives and false negatives. Next, they moved them into the "Post Collections" panel. Then they updated the AutoModerator configuration to resolve the collected posts on the Post Collections panel.

More specifically, we observed two patterns in collecting problematic posts (usually actual false positives and false negatives) in "Post Collections" and in updating the rules. The first pattern was collecting multiple problematic posts at once and updating the rules referring all of them together. This pattern was what we had expected before the user study. While six participants (P1, P5, P6, P8, P9, P10) showed this pattern, three participants (P6, P8, and P9) failed to use this pattern because they could not find a breakthrough configuration to resolve the collected false positives and false negatives. P6's configuration was filtering nearly 50% of posts in Task A, and he said *"I can't think of a better word specifically, that would be good to exclude or add to the string to check."* while checking the false positives in the Post Collections. The second pattern was different in that the posts were collected one by one in the Post Collections panel, followed by a rule update after each collection. This resulted in fine-tuning the rules to resolve each and every post collected. Two participants (P4, P7) introduced this process at first and the other two participants (P1, P5) who first showed the first pattern later adopted the second pattern. All participants showing the second pattern presented a common strategy to update the rules, which is writing a filter with a large number of keywords at first and then adding white-list keywords to exclude false positives. An exceptional pattern was observed from P2, where she did not use the Post Collections panel at all and directly reflected the false positives she found from the "Possible false alarms" panel. P2 reported that it was cumbersome to move posts to the Post Collections because she was able to just quickly deal with the false positives she found without having to move them.

Six participants (P1, P5, P7, P8, P9, P10) used the "Configuration Analysis" panel as an extra supporting tool to understand how the rules are working. Four of them (P1, P5, P9, P10) refer to this feature to quickly find which part of

Condition	Task	Sandbox	Collections	FP & FN	Analysis	System
Experienced	Task A	4.8(1.8)	**5.2(1.9)**	**6.0(0.7)**	5.2(0.8)	**6.2(0.8)**
	Task B	4.6(1.8)	**6.2(0.8)**	5.4(1.1)	5.2(1.1)	
Novice	Task A	5.0(1.6)	4.6(1.6)	**5.8(0.8)**	5.0(1.6)	5.4(2.1)
	Task B	**6.0(1.5)**	5.0(0.9)	5.4(0.4)	5.0(2.0)	
	Total	5.1(1.6)	5.2(1.6)	**5.7(0.9)**	5.1(1.4)	

Table 3. Average usefulness scores (and standard deviation) of each feature in ModSandbox

the rules are filtering the false positives and false negatives in "Post Collections". The other two (P7 and P8) checked the number of posts that each rule and keyword affected. They first checked whether their rules were catching too many posts or not by looking at the ratio bar in the sandbox panel and used the Configuration Analysis function to remove the relevant part of the rules. The rest of the participants rarely used this function.

Rule-making pattern in ModSandbox. Using features in ModSandbox, participants refined the rules written in the baseline system multiple times. We note that we emphasized in each study session that the rules written in the baseline system should be in the form of their best attempt before moving to use ModSandbox. Figure 9 describes how experienced moderators changed their rules over time. Compared to the baseline system (Figure 8), the number of filtered posts tends to decrease over time. They also change the rule more frequently. It implies that they try to fine-tune the configuration with more attempts. We observed that P7, P8, and P10 added new keywords in their rules because they wanted to make more complex configurations with multiple checks and rules. As a result, the rules updated using ModSandbox tend to be more complex than the baseline system (Table 2).

5.4.2 Post survey results. After the user study, we asked all participants how useful each feature was and how they think they can be improved. We summarize the responses, highlighting the main strength of each feature.

Feature 1. A Sandbox Environment: Valuable to see what posts are being filtered. Six participants (P2, P3, P4, P6, P8, P9) responded that a sandbox environment was valuable because they could see what posts are being filtered in real-time. P1 said that he gave a high score for this function because it can show the results of AutoModerator applied to the community without affecting the community. However, three participants (P4, P7, P9) said that the Sandbox UI showed so much data compared to Feature 3: View Possible Misses and False Alarm, that they did not like it. P7 said *"There are a lot posts shown at a time, which makes it less useful compared to the features with fewer posts shown."*

Feature 2. Post Collections Panel: More useful to participants familiar with configuring AutoModerator. Six participants(P1, P2, P6, P7, P8, P10) mentioned that seeing the posts manually gathered in the Post Collections panel was useful for the user study task. Specifically, P6, P7, and P8 noted their usefulness in finding proper rules. P6 reported *"It was great to actually see which posts are false negatives and false positives so that it was easier to look for keywords that are more relevant to the current topic."* However, Three novice moderators (P3, P6, P9) pointed out they are difficult to use and gave lower usefulness scores to this feature (Table 3). P9 said *"It was useful in that is showed me how the keywords were being used but it left me wondering how to apply this."*

Feature 3. View Possible Misses and False Alarms: The most useful feature for everyone, but only works well if posts have semantic similarities. Five participants (P1, P4, P5, P8, P10) liked this feature because it allowed them to grasp probable false positives and false negatives and quickly find actual false positive and negative posts. P10 mentioned *"The possible*

misses, false alarms was very helpful in showing what things I missed with my filter. It definitely saved me tons of time of scrolling through matches to find bad ones." However, P2, P3, and P10 felt that possible misses are less accurate in Task B. P3 wrote *"This feature is so convenient, but I think there were many articles in the Possible Misses that did not seem to be included in the task"*. Interestingly, P7 doubted the accuracy of the algorithm *"I'm unsure how good the algorithm is and I'd be afraid that focusing on these will miss important posts"*. They evaluated this function as less useful in Task B, but as the most useful function overall (Table 3). We note that this feature was indeed less accurate in Task B because each post mentioning COVID-19 had very different semantic and context compared to Task A. Targeted posts in Task A shared similar topics, but targeted posts in Task B had varying topics.

Feature 4. Configuration Analysis and Highlights: More useful with more complex rules. "Configuration Analysis" panel helped Four experienced moderators (P5, P7, P8, P10) to analyze the code and determine which rules or words are good and bad for the task. P5 answered that it is helpful to see the impact each code is having on the filtered results, helping them to eliminate keywords that were yielding too many false results. Two experienced moderators (P1, P8) stated that they could understand how the rules are working by themselves and they did not feel the need to use the panel much. Interestingly, P1 suggested a novel way to utilize what is seen in the Configuration Analysis panel. He pointed out that it is easily readable data that could be presented to other moderators as evidence to discuss the flaws and strengths of each rule. Four novice moderators (P2, P3, P4, P9) preferred "Highlights" because it helps notice where the keywords in the rules are. P4 felt confident that she could identify why certain keywords were filtered or not.

Rule-making strategies in ModSandbox. Participants elaborated the rule-making strategies they used while using ModSandbox. Four participants (P2, P4, P6, P8) first created a rule with a large list of keywords to catch targeted posts and update the rules to reduce the false positives. To be specific, P2, P4, and P8 added some white-list keyword filters to exclude the false positives. P8 described *"First I scanned the posts and the task for keywords that might be able to match what I want. I then checked the false positives to find additional keywords that I could add to the rules to reduce the amount of false positives"*. P6 followed the same strategy, but he confessed that the configuration became so muddled that he was getting too many false positives and regretted that he should have thought a lot more different than he had at the beginning in trying to filter his words. P7 introduced an impressive strategy. He first tried to think of a simple algorithm that can catch or reject targeted posts in this head, e.g., find all posts including word X and word Y, but not any posts including word Z. Then he thought of keywords that fulfill that logic.

Feedback to improve ModSandbox. Three moderators (P1, P4, P7) provided feedback to improve ModSandbox through the post-survey. P1 and P4 mentioned that the UI could be more simplified so that novice or casual moderators could also easily use it. P4 and P7 suggested analyzing word frequency in the "Post Collections" panel so that the most frequent words can be used as recommended keywords when writing keyword-based automated rules.

6 DISCUSSION

In this work, we investigated challenges that online content moderators encountered when using automated moderation bots and presented a novel approach to help them rapidly test the impact of these bots without affecting their actual communities to improve the performance of those bots. Below we discuss the impact of intelligent NLP algorithms that help find false positives and false negatives, the potential impact of recommending concrete methods on how to update automated rules, supporting efficient collaboration between moderators, reducing emotional labors for online content moderators through using ModSandbox, and ModSandbox being a practical solution for other platforms beyond Reddit.

6.1 The Impact of Intelligent Algorithms on Finding False Positives and False Negatives

The accuracy of the algorithms for detecting possible false alarms and misses had a significant impact on participants' trust and perceived usefulness of the system. During Task B of our user study, three participants commented that possible false alarms and misses did not seem accurate. P7 doubted the performance of the algorithm: *"I'm unsure how good the algorithm is and I'd be afraid that focusing on these will miss important posts."* We believe that our algorithm was less accurate for Task B because it calculated the semantic similarities in the level of sentences, not keywords. It worked better in Task A because it asked to detect posts in a similar context that asks about jobs, but Task B was about detecting posts with keywords related to COVID-19. These posts could have any topic spanning from talking about the impact of anti-vaccine protests in the job market to having to work remotely due to quarantine. While Task B represents a typical ban to block the recent controversial source to avoid off-topic discussion, our algorithm, Universal Sentence Encoder [6], is not suitable for this kind of task where relevant keywords could be used in different contexts. Different algorithms may be more suitable for different keyword-filtering scenarios and moderation objectives algorithms. For the current system, we only use a single algorithm, but we could improve the system to support alternative algorithms for detecting false alarms and misses, and let moderators compare the performance between them and apply what works best for them.

Another way to improve this feature is expanding the range of imported data. ModSandbox extracts the possible misses and false alarms from only the posts on one subreddit. The system can potentially use posts from multiple, similar subreddits that share similar norms [9]. A more significant number of post data can help moderators make a concrete and preemptive configuration because they can provide various examples that reflect prospective behavior from similar communities [5].

6.2 Recommending Concrete Ideas on How to Update the Automated Rules

Showing false positives and false negatives is not enough for moderators to find a pattern to avoid them. During the user study, three moderators (P6, P8, P9) found it hard to extract meaningful patterns from multiple posts in the "Post Collections" panel. They lacked ideas to update the rules using these patterns because they could not identify adequate keywords. As a solution, we can leverage the Post Collections panel to suggest concrete directions to improve the configuration. In the study, two participants (P4, P7) suggested showing frequently occurring keywords and inverse frequency analysis, which is a method to measure how much information each word provides, to help find useful keywords to improve the rules based on the collected posts. Furthermore, ModSandbox can potentially suggest a single regular expression that detects those useful keywords.

Rule updating patterns observed in the user study can be clues to designing recommendations for rule-making. For example, some participants added keywords they found in the false positive examples as white-list keywords. Furthermore, P10 started with a single check with a list of keywords and then added an additional normal check or reverse check, which is a condition that the post must not meet. These structured patterns can become a framework to form a better AutoModerator configuration.

Guiding moderators to make informed updates to their AutoModerator configuration is a promising next step. The system can potentially recommend effective rules based on keyword extraction results and rule updating patterns. However, such data-driven recommendations may sometimes suggest rules that humans cannot understand or express as recommendation algorithms do. Thus, the system should provide different recommendation options to choose from so that moderators can build complex enough but understandable rules.

6.3 Facilitating Learning and Collaboration Between Moderators

In the user study, P1, who moderates a high-traffic subreddit, said ModSandbox *"would not only allow for refinement of rules, but presentation thereof"*. P1 meant that one could use ModSandbox to demonstrate the expected results of AutoModerator configurations during discussions within peer moderators, which they carefully go through before initiating any moderation decision. P1 suggested that such usage of ModSandbox can help casual moderators be more involved in the configuration process. A previous study [18] showed that only a few moderators actively configure AutoModerator because of its difficulty in learning how to use it. Thus they suggested that an automated system could be designed to make it easier for moderators to understand how to use it. Moderation supporting tools that can visualize moderation rules and their results, such as ModSandbox, can be a promising solution to support many non-tech savvy moderators to engage in automated tools. In addition, ModSandbox can support them to learn how to use a regular expression in the configuration by testing it in the sandbox environment. We expect this line of work to lower the barriers to novice moderators by providing a lightweight learning opportunity.

Furthermore, ModSandbox has the potential to serve a team of moderators in their distributed decision-making scenarios. We can extend ModSandbox to support multiple moderators to collaboratively author and discuss moderation rules. We could even give these capabilities to community users, increasing moderation transparency and awareness. Recent studies([20, 40, 50]) have introduced software infrastructures and strategies to support distributed governance for online communities. ModSandbox contributes a special-purpose software infrastructure and governance layer for algorithmic moderation to this thread of research.

6.4 Reducing Emotional Labor in Setting Up Automated Moderation Tools

The feature "View Possible Misses and False Alarms" can help reduce emotional labor for moderators when setting up automated tools. In our study, this feature strongly helped participants identify bugs or undesired actions of AutoModerator configuration without skimming through all the posts. P10 mentioned that ModSandbox saved time in finding problematic posts compared to scrolling through a large number of community posts.

Another benefit of using the "View Possible Misses and False Alarms" feature is that the moderators can avoid being exposed to toxic and harassing posts during moderation. While the typical moderation process requires emotional labor to the moderators because they are exposed to these toxic posts while skimming through posts [10, 26, 36, 38], using the View Possible Misses and False Alarms feature create a separate space for the moderators to focus only on the posts related to the current moderation task. Facebook recently built an AI-supported moderation system to reduce the post that paid moderators should review by automatically excluding obviously harmful content and sorting ambiguous content first [47]. In a similar vein, ModSandbox also provides algorithmic support to reduce the number of posts moderators need to review. That is, our study suggested approaches that can help reduce the emotional labor of online content moderators when creating automated rules.

6.5 Providing Practical Moderation Solution to Online Community Platforms Beyond Reddit

Beyond Reddit, the design implication of ModSandbox can be applied to various social platforms because it supports keyword-based filtering that most platforms have. The keyword filter is being used in many social platforms like Facebook [3], YouTube [4], and Twitter [5]. Our approach can work on those platforms as well because it only requires a

[3]https://www.facebook.com/formedia/blog/moderating-your-facebook-page
[4]https://support.google.com/youtube/answer/9483359
[5]https://help.twitter.com/en/using-twitter/advanced-twitter-mute-options

filtering algorithm and a set of APIs that retrieves existing posts from a platform. It enables to package ModSandbox in the form of browser add-on like Toolbox for Reddit [6], which is one of the popular tools for Reddit moderators [18].

7 LIMITATIONS

There are several limitations of the research that should be acknowledged. As our system was built on top of AutoModerator, an inherent barrier to entry is that moderators should learn to write rules in YAML. While out of scope in this research, the difficulty of learning AutoModerator could constrain the potential user pool. In our study, we gave tutorials and quizzes to ensure participants could author AutoModerator rules and everyone successfully performed the study tasks. But we acknowledge that this is not likely to be expected in practice, thereby deterring moderators from using AutoModerator altogether. Future work can consider adding support for making YAML and even regex authoring easier, which can be a complementary feature to what ModSandbox offers.

The user study did not fully simulate the AutoModerator configuration in the actual moderation environment. While participants are used to moderating their community, we provided posts from a different community they had not managed before. They may have a limited understanding of the posts used in the study. Furthermore, while study participants reviewed only the posts on the live community, moderators usually read additional content like AutoModerator logs or a spam queue as well. Those differences can affect the user experience in the AutoModerator configuration task. After all, future research needs to conduct a live deployment experiment on how tools work in a real-world moderation environment.

ModSandbox only focuses on supporting a specific type of moderation: content-based keyword filtering by Reddit AutoModerator. However, the original AutoModerator can filter posts by various metadata like the age of the accounts and length of the body [7]. Furthermore, moderators in other subreddits or community platforms have used different third-party or built-in tools to automate complex tasks such as deleting old posts [25, 33, 43]. What ModSandbox is supporting still accounts for a large portion of automated moderation in an online space [35]; however, ModSandbox has room for improving its UI and algorithm to support more diverse moderation practices. In future work, researchers can augment ModSandbox to support other general tools for various platforms.

8 CONCLUSION

This paper proposes ModSandbox, a novel testing system for automated content moderation bots, which enables human moderators to rapidly verify the effect of the bots without affecting their actual communities. We found the need and design considerations of ModSandbox through a series of surveys and interviews with Reddit moderators. Our user study with expert moderators from various platforms demonstrates that ModSandbox not only helps configure automated rules that reflect detailed intentions of the moderators but also helps detect possible false positives and negatives to rapidly update the rules to minimize them. The update through ModSandbox can happen in a few hours, while the conventional method took days or even weeks. Potential extended use cases of ModSandbox include support collaboration between moderators by sharing the results from the system and supporting novice moderators to learn how to be an expert in using automated moderation bots.

REFERENCES

[1] Iris Birman. 2018. Moderation in different communities on Reddit–A qualitative analysis study. (2018).

[6] https://www.reddit.com/r/toolbox/
[7] https://www.reddit.com/wiki/automoderator/full-documentation

[2] Virginia Braun and Victoria Clarke. 2012. Thematic analysis. (2012).

[3] Michael Brooks, Saleema Amershi, Bongshin Lee, Steven M Drucker, Ashish Kapoor, and Patrice Simard. 2015. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 105–112.

[4] Twitter Help Center. 2021. *Hateful conduct policy.* Retrieved 2021 from https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy

[5] Damon Centola. 2010. The spread of behavior in an online social network experiment. *science* 329, 5996 (2010), 1194–1197.

[6] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 169–174.

[7] Stevie Chancellor, Jessica Annette Pater, Trustin Clear, Eric Gilbert, and Munmun De Choudhury. 2016. #thyghgapp: Instagram content moderation and lexical variation in pro-eating disorder communities. In *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*. 1201–1213.

[8] Eshwar Chandrasekharan, Chaitrali Gandhi, Matthew Wortley Mustelier, and Eric Gilbert. 2019. Crossmod: A cross-community learning-based system to assist reddit moderators. *Proceedings of the ACM on human-computer interaction* 3, CSCW (2019), 1–30.

[9] Eshwar Chandrasekharan, Mattia Samory, Shagun Jhaver, Hunter Charvat, Amy Bruckman, Cliff Lampe, Jacob Eisenstein, and Eric Gilbert. 2018. The internet's hidden rules: An empirical study of Reddit norm violations at micro, meso, and macro Scales. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–25.

[10] Bryan Dosono and Bryan Semaan. 2019. Moderation practices as emotional labor in sustaining online communities: The case of AAPI identity work on Reddit. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.

[11] Siamak Faridani, Ephrat Bitton, Kimiko Ryokai, and Ken Goldberg. 2010. Opinion space: a scalable tool for browsing online comments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1175–1184.

[12] R Stuart Geiger. 2016. Bot-based collective blocklists in Twitter: the counterpublic moderation of harassment in a networked public space. *Information, Communication & Society* 19, 6 (2016), 787–803.

[13] R Stuart Geiger and Aaron Halfaker. 2013. When the levee breaks: without bots, what happens to Wikipedia's quality control processes?. In *Proceedings of the 9th International Symposium on Open Collaboration*. 1–6.

[14] R Stuart Geiger and David Ribes. 2010. The work of sustaining order in Wikipedia: The banning of a vandal. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. 117–126.

[15] Daniel J Hruschka, Deborah Schwartz, Daphne Cobb St. John, Erin Picone-Decaro, Richard A Jenkins, and James W Carey. 2004. Reliability in coding open-ended data: Lessons learned from HIV behavioral research. *Field methods* 16, 3 (2004), 307–331.

[16] Fernando Alfonso III. 2014. *Reddit's r/technology has apparently been blocking Tesla links for months.* Retrieved 2021 from https://www.dailydot.com/unclick/reddit-technology-tesla-ban/

[17] Shagun Jhaver, Darren Scott Appling, Eric Gilbert, and Amy Bruckman. 2019. " Did You Suspect the Post Would be Removed?" Understanding User Reactions to Content Removals on Reddit. *Proceedings of the ACM on human-computer interaction* 3, CSCW (2019), 1–33.

[18] Shagun Jhaver, Iris Birman, Eric Gilbert, and Amy Bruckman. 2019. Human-machine collaboration for content regulation: The case of Reddit Automoderator. *ACM Transactions on Computer-Human Interaction (TOCHI)* 26, 5 (2019), 1–35.

[19] Shagun Jhaver, Amy Bruckman, and Eric Gilbert. 2019. Does transparency in moderation really matter? User behavior after content removal explanations on reddit. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–27.

[20] Shagun Jhaver, Seth Frey, and Amy Zhang. 2021. Designing for Multiple Centers of Power: A Taxonomy of Multi-level Governance in Online Social Platforms. *arXiv preprint arXiv:2108.12529* (2021).

[21] Shagun Jhaver, Sucheta Ghoshal, Amy Bruckman, and Eric Gilbert. 2018. Online harassment and content moderation: The case of blocklists. *ACM Transactions on Computer-Human Interaction (TOCHI)* 25, 2 (2018), 1–33.

[22] Jialun Aaron Jiang, Charles Kiene, Skyler Middler, Jed R Brubaker, and Casey Fiesler. 2019. Moderation challenges in voice-based online communities on discord. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–23.

[23] Prerna Juneja, Deepika Rama Subramanian, and Tanushree Mitra. 2020. Through the Looking Glass: Study of Transparency in Reddit's Moderation Practices. *Proceedings of the ACM on Human-Computer Interaction* 4, GROUP (2020), 1–35.

[24] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. 2010. Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1343–1352.

[25] Charles Kiene and Benjamin Mako Hill. 2020. Who Uses Bots? A Statistical Analysis of Bot Usage in Moderation Teams. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–8.

[26] Charles Kiene, Andrés Monroy-Hernández, and Benjamin Mako Hill. 2016. Surviving an" Eternal September" How an Online Community Managed a Surge of Newcomers. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1152–1156.

[27] Sara Kiesler, Robert Kraut, Paul Resnick, and Aniket Kittur. 2012. Regulating behavior in online communities. *Building successful online communities: Evidence-based social design* (2012), 125–178.

[28] Kate Klonick. 2017. The new governors: The people, rules, and processes governing online speech. *Harv. L. Rev.* 131 (2017), 1598.

[29] Travis Kriplean, Jonathan Morgan, Deen Freelon, Alan Borning, and Lance Bennett. 2012. Supporting reflective public thought with considerit. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. 265–274.

[30] Travis Kriplean, Michael Toomim, Jonathan Morgan, Alan Borning, and Andrew Ko. 2012. Is this what you meant? Promoting listening on the web with reflect. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1559–1568.

[31] PJS Kumar, Polagani Rama Devi, N Raghavendra Sai, S Sai Kumar, and Tharini Benarji. 2021. Battling Fake News: A Survey on Mitigation Techniques and Identification. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 829–835.

[32] Claudia Claudia Wai Yu Lo. 2018. *When all you have is a banhammer: the social and communicative work of Volunteer moderators*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[33] Kiel Long, John Vines, Selina Sutton, Phillip Brooker, Tom Feltwell, Ben Kirman, Julie Barnett, and Shaun Lawson. 2017. " Could You Define That in Bot Terms"? Requesting, Creating and Using Bots on Reddit. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3488–3500.

[34] Inc. Online Moderation. 2019. *Online Moderation*. Retrieved 2021 from https://www.onlinemoderation.com/human-moderators-versus-technology/

[35] Reddit. 2020. Transparency Report 2020. https://www.redditinc.com/policies/transparency-report-2020

[36] Sarah T Roberts. 2016. Commercial content moderation: Digital laborers' dirty work. (2016).

[37] Sarah T Roberts. 2018. Digital detritus:'Error'and the logic of opacity in social media content moderation. *First Monday* (2018).

[38] Sarah T Roberts. 2019. *Behind the screen*. Yale University Press.

[39] Koustuv Saha, Eshwar Chandrasekharan, and Munmun De Choudhury. 2019. Prevalence and psychological effects of hateful speech in online college communities. In *Proceedings of the 10th ACM conference on web science*. 255–264.

[40] Nathan Schneider, Primavera De Filippi, Seth Frey, Joshua Z Tan, and Amy X Zhang. 2021. Modular Politics: Toward a Governance Layer for Online Communities. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–26.

[41] Joseph Seering. 2020. Reconsidering Self-Moderation: the Role of Research in Supporting Community-Based Models for Online Content Moderation. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 1–28.

[42] Joseph Seering, Robert Kraut, and Laura Dabbish. 2017. Shaping pro and anti-social behavior on twitch through moderation and example-setting. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*. 111–125.

[43] Joseph Seering, Tony Wang, Jina Yoon, and Geoff Kaufman. 2019. Moderator engagement and community development in the age of algorithms. *New Media & Society* 21, 7 (2019), 1417–1443.

[44] Burr Settles. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. 1467–1478.

[45] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S Tan. 2009. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1283–1292.

[46] Sahaj Vaidya, Jie Cai, Soumyadeep Basu, Azadeh Naderi, Donghee Yvette Wohn, and Aritra Dasgupta. 2021. Conceptualizing Visual Analytic Interventions for Content Moderation. In *2021 IEEE Visualization Conference (VIS)*. IEEE, 191–195.

[47] James Vincent. 2020. Facebook is now using AI to sort content for quicker moderation. *The Verge* (Nov. 2020). https://www.theverge.com/2020/11/13/21562596/facebook-ai-moderation

[48] Wikipedia. 2021. *Wikipedia: Wikipedia is not a forum*. Retrieved 2021 from https://en.wikipedia.org/wiki/Wikipedia:Wikipedia_is_not_a_forum

[49] Donghee Yvette Wohn. 2019. Volunteer moderators in twitch micro communities: How they get involved, the roles they play, and the emotional labor they experience. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.

[50] Amy X Zhang, Grant Hugh, and Michael S Bernstein. 2020. PolicyKit: Building Governance in Online Communities. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 365–378.

A SURVEY QUESTIONNAIRE

We asked our respondents the following questions in the both surveys:

A.1 Questions in the first survey

1. How long have you been a moderator in Reddit?

   _____

2. How many subreddits do you participate in as an active moderator?

   _____

3. Do you know how to configure AutoModerator?
   a. Yes, I'm an expert
   b. Yes, I'm not an expert but I know enough to use in my own sub
   c. Well, I think I know a little bit

d. No, I don't know at all

4. Do you configure AutoModerator by yourself?
   a. Yes, most of the time
   b. Yes, occasionally
   c. Well, I've done it once or twice
   d. No, never
5. Please tell us about what things are done using AutoModerator in your subreddit.

---

6. If there were/are anything hindering you from "learning" how to configure AutoModerator, what is the reason? Please select all that apply.
   a. I don't know how to program, so I don't know where to start
   b. I can't practice because I'm afraid I'll mess up with the current AutoModerator configurations in my subreddit
   c. I don't think there's any (or insufficient) tools or methods to practice AutoModulator configurations
   d. The documentations on how to use AutoModerator are poor
   e. I don't have anyone to teach me how to use AutoModerator
   f. I don't feel the need to learn since other moderators can configure AutoModerator for my subreddit
   g. I don't feel the need to learn since my subreddit can be moderated without AutoModerator
7. Have you ever felt at a disadvantage because you don't know how to configure AutoModerator? (e.g., you can't join a sub as a mod, your authorities are limited, mod discussions are made without you, etc)
   a. Most of the time
   b. Sometimes
   c. Seldom (rarely)
   d. Not at all
8. If you read a configuration that is written by someone else, do you understand the exact actions the AutoModerator would take?
   a. Most of the time
   b. Sometimes
   c. Seldom (rarely)
   d. Not at all
9. If you have ever thought that you want to / need to understand how AutoModerator works, why did you think so?

---

10. Have you ever thought that, because very few moderators know how AutoModerator works, the moderation process is less transparent to other moderators?
    a. Most of the time
    b. Sometimes
    c. Seldom (rarely)
    d. Not at all
11. Have you ever felt that the mods who know how to configure AutoModerator have too much work to do, causing imbalanced workload between moderators?

a. Most of the time
b. Sometimes
c. Seldom (rarely)
d. Not at all

12. Have you ever thought it would be helpful if there are more moderators who can configure AutoModerator?
    a. Most of the time
    b. Sometimes
    c. Seldom (rarely)
    d. Not at all
13. Have you ever felt that the mods who know how to configure AutoModerator have more power over other moderators in the moderation decision/process?
    a. Most of the time
    b. Sometimes
    c. Seldom (rarely)
    d. Not at all
14. Are there any specific episodes you can share regarding the previous questions?

    _____

15. Has applying changes to AutoModerator configuration ever brought unexpected results or side effects on your subreddit?
    a. Most of the time
    b. Sometimes
    c. Seldom (rarely)
    d. Not at all
16. If you answered yes in the previous question, what were the circumstances?

    _____

17. Have you ever felt frustrated by the fact that no mistake is allowed when applying a new Automod configuration to your subreddit, i.e., the configuration affects the subreddit immediately?
    a. Most of the time
    b. Sometimes
    c. Seldom (rarely)
    d. Not at all
18. What things do you carefully check before applying a new AutoModerator configuration to your subreddit?

    _____

19. If you use any form of virtual sandbox to test AutoModerator configurations before you apply it to your subreddit, please tell us about your practice.

    _____

20. If there's a virtual sandbox webpage where you can freely test and explore AutoModerator configurations, which things would you like to test? (e.g., if the configuration works as intended, if the users would like the change, if the layout looks okay, etc.)

    _____

A.2 Questions in the second survey

1. What is your gender?
   a. Female
   b. Male
   c. Prefer not to say
   d. Other
2. What is your age category?
   a. 18-24
   b. 25-34
   c. 35-44
   d. 45-54
   e. 55-64
   f. 65 and above
3. What is the highest degree or level of education you have completed? If currently enrolled, select the highest degree received.
   a. Some high school, no diploma, and below
   b. High School
   c. Bachelor's Degree
   d. Master's Degree
   e. Ph.D. or higher
4. How long have you been a moderator in Reddit?

   _____
5. How many subreddits do you participate in as an active moderator?

   _____
6. Do you know how to configure AutoModerator?
   a. Yes, I'm an expert
   b. Yes, I'm not an expert but I know enough to use in my own sub
   c. Well, I think I know a little bit
   d. No, I don't know at all
7. Do you configure AutoModerator by yourself?
   a. Yes, most of the time
   b. Yes, occasionally
   c. Well, I've done it once or twice
   d. No, never
8. Do your subreddits use AutoModerator? If so, please tell us about what things are done using AutoModerator in your subreddits.

   _____
9. Do you use AutoModerator to handle posts or comments that contain particular keywords? If so, could you briefly share any example cases?

   _____

10. Have you ever experienced that "innocent" posts were affected by Automod keyword filtering rules (i.e., false positives)? If so, what were the circumstances?

_____

11. If you don't mind participating in a further in-depth interview on "how to set up AutoModerator", please leave an e-mail address below where we can contact you for the interview. The participants will be compensated with an Amazon gift card (expected to be $30) upon completion of the interview, which will last 30-60 minutes. The email information you provide here will not be shared outside the research group.

_____