Moderation Suite for Reddit

Himanshu Singh

May 24, 2023

Contents

List of Figures

Listings

Abstract

This project aims to reduce the workload of Reddit moderators, who have the responsibility of controlling public discourse in their respective subreddits. The problems faced by them were identified by checking various meta subreddits, and relating them with the established results. We realised most of these could be automatically detected and resolved by performing simple computations and communicating with Reddit API. We combined these protocols with an easy-to-use GUI to make a reliable moderation suite.

1 Introduction

Reddit is a pseudonymous website comprising of over 3.4 million user-run communities, called "subreddits" [4]. Registered users can make submissions and comments to any public subreddit, thereby making it accessible to all the subreddit members, unless it is stuck under a filter. [12]

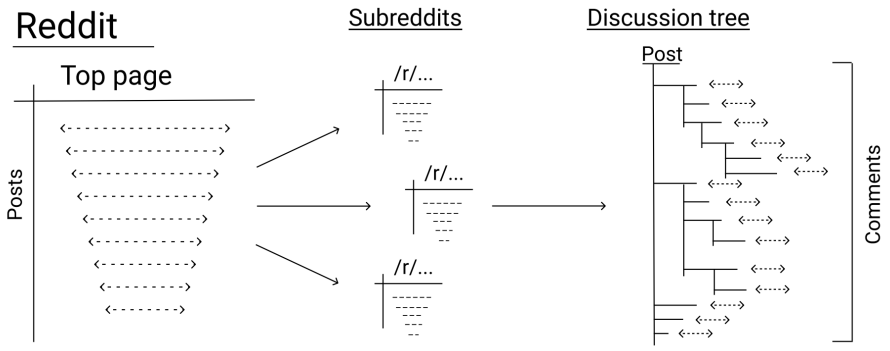

Figure 1: The schematic structure of Reddit [10]

Reddit takes a distributed approach for moderation, with the admins enforcing platform-wide rules, and the "voluntary" moderators deciding on the subreddit specific rules. [7] [8]. Subreddit moderation in itself is a highly subjective task, that often requires discussion amongst the various moderators prior to taking any sort of action. [9] It is thus essential to take into account input from users before automating any action.

Reddit allows "crosslinking" of posts in other communities, allowing an easy influx of users from one subreddit to another. Trogus et al. found that 27.8% crosslinks led to negative interactions, 78% of which were caused by top 1% communities. [17] This can be detrimental for the survival of smaller communities, making them vulnerable to manipulation of submission scores, which directly affects their visibility. [2] Another study, by Datta and Adar noted that 82% of controversial authors have only a single

“anti-social home”. [3] This provides an easy tracking mechanism for users involved in negative interactions.

This suggests that identification of mobilizing communities and subsequent blocking of cross-community interaction can curb majority of conflicts. However, inbuilt moderation tools like AutoModerator and Crowd Control go for simple blocking mechanisms, that does not take into account the nature of involved subreddits. [5] [6] Hence, a new blocking mechanism needs to be explored.

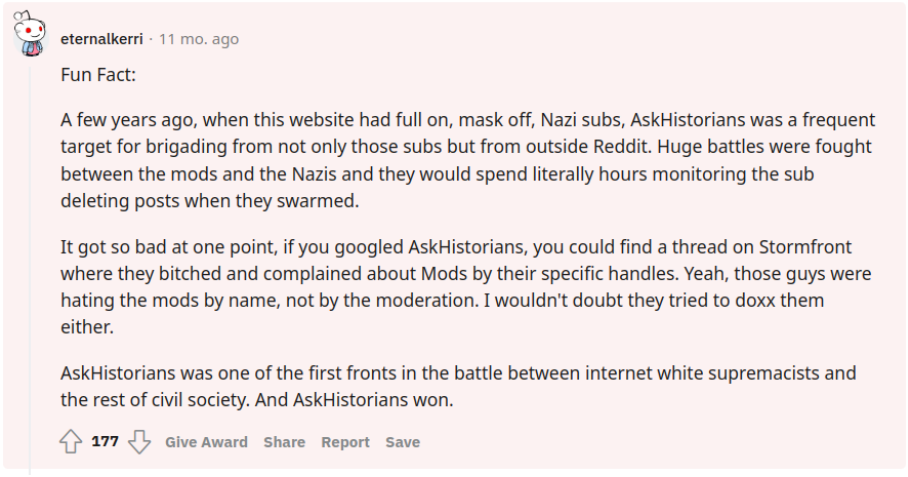

Figure 2: u/eternalkerri describes the targetting of r/AskHistorians [13]

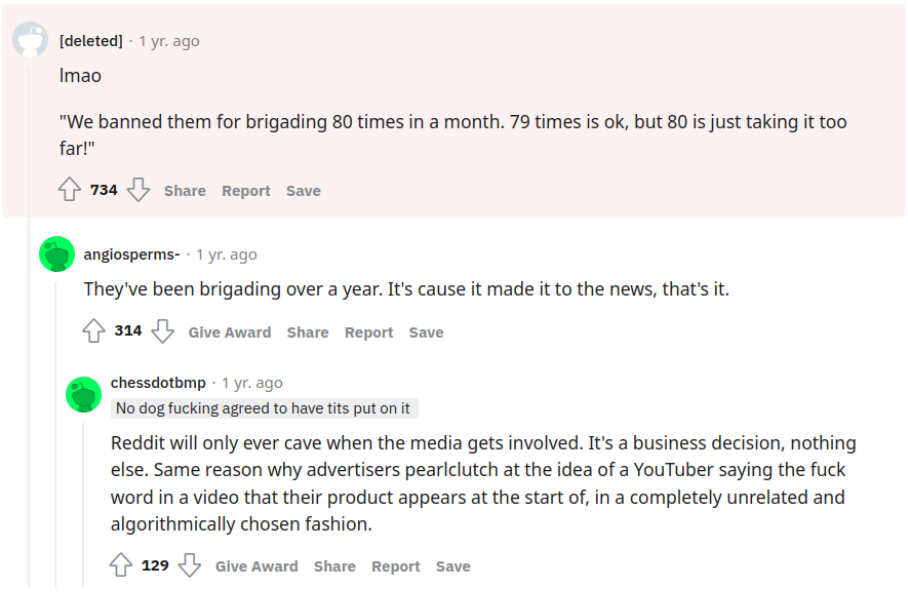

Figure 3: r/SubredditDrama members discuss the ban of r/NoNewNormal [14]

Figure 4: u/NomaiTraveler's suggested notion for "brigading" [14]

2 Methodology

We start by exploring the subreddits with the highest overlap. From here on, we will be calling the number of shared users between two subreddits as simply "overlap", and the probability of our subreddit's member commenting in another subreddit as "participation probability". [Jump to 'Implementation of User Overlap' on page 6] The latter having the advantage of detecting smaller but better overlapping subreddits more accurately.

Based on the results computed, the moderator is asked to flag the subreddits that she deems to be hostile. Additionally, the moderator can flag a list of keywords that she deems to be recurrent in negative interactions, and characterize what she deems to be a "throwaway" account. [18] Then, the list of users to be banned is generated with respect to various tolerance limits, to avoid false positives. [Jump to 'Implementation of Prune Members' on page 6]

A complementary feature is provided to purge all the submissions and comments left by users who were either banned earlier or were allowed to pass through by our tolerance limits. Moreover, this protocol allows the moderator to flag a list of domains on similar lines to keywords. [Jump to 'Implementation of Purge Content' on page 7]

Lastly, the moderator is provided with a set of charts to help her understand the changes in her subreddit over a period of time. This would help her determine the area and extent of action needed for her subreddit. This includes the number of active users, submissions, comments, scores, type of content, usage of flairs and frequency of edits over a period of time. [Jump to 'Implementation of Subreddit Trends' on page 8]

3 Implementation

Around December 25th 2021, the creator of u/Flair_Helper bot, u/Blank-Cheque reported that bots moderating more than 500 subreddits were being limited. [11] Hence, to implement the features discussed in the previous section, we will be looking at an application that can be run locally by any number of users. The application is designed

using GTK widgets for cross-platform compatibility, and written in Python to leverage the benefits offered by PRAW library. [1] [16] To avoid theming inconsistency, we will using libadwaita's ApplicationWindow as the top-level container. [15]

Listing 1: Computation of User Overlap [1]

```
1  userlist = []
2  for item in reddit.subreddit(home).new(limit = lim/10):
3      try:
4          author = item.author.name
5          if author not in userlist:
6              if author != "[deleted]":
7                  userlist.append(author)
8      except:
9          continue
10 for item in reddit.subreddit(home).comments(limit = 9*lim/10):
11     try:
12         author = item.author.name
13         if author not in userlist:
14             if author != "[deleted]":
15                 userlist.append(author)
16     except:
17         continue
18
19 overlap = {}
20 for user in userlist:
21     try:
22         profile = {}
23         for item in reddit.redditor(user).new(limit = ulim):
24             if item.subreddit.display_name.startswith("u_") or (item.subreddit.display_name
                   == home):
25                 continue
26             profile[item.subreddit.display_name] = 1
27         for sub in profile:
28             if sub in overlap:
29                 overlap[sub] += 1
30             else:
31                 overlap[sub] = 1
32
33 prob = {}
34 scale = 100000000/(ulim*len(userlist))
35 for sub in overlap:
36     subcount = reddit.subreddit(sub).subscribers
37     prob[sub] = round(overlap[sub]*scale/subcount, 2)
```

Listing 2: Implementation of Prune Members

```
1  banlist = []
2  currtime = time.time()
3  for user in userlist:
4      profile = reddit.redditor(user)
5      if profile.link_karma < lklim:
6          banlist.append(user)
```

[1] For the sake of keeping the listing concise, we have omitted the initialization of several objects, and passing of parameters. The complete implementation is publicly available at https://github.com/ihsingh2/ms4r.

```
            continue
        if profile.comment_karma < cklim:
            banlist.append(user)
            continue
        if (currtime − profile.created_utc) < agelim:
            banlist.append(user)
            continue

        count = 0
        history = list(profile.new(limit = 100))
        threshold = min(1, int(len(history)/10))
        for item in history:
            try:
                if count >= threshold:
                    banlist.append(user)
                    break
                if item.subreddit.display_name in sblacklist:
                    count += 1
                if isinstance(item, praw.models.Comment):
                    if any(word in item.body for word in wblacklist):
                        count += 1
                elif isinstance(item, praw.models.Submission):
                    if any(word in item.title for word in wblacklist):
                        count += 1
                    if item.is_self:
                        if any(word in item.selftext for word in wblacklist):
                            count += 1
            except:
                continue
```

Listing 3: Implementation of Purge Content

```
for item in reddit.subreddit(home).comments():
    if days != 0:
        if item.created_utc > rmvafter:
            item.mod.remove()
            continue
    if rmvban:
        if item.author != None:
            if item.author.name != ”[deleted]”:
                if any(reddit.subreddit(home).banned(item.author.name)):
                    item.mod.remove()
                    continue
    if any(word in item.body for word in wblacklist):
        item.mod.remove()

for item in reddit.subreddit(home).new():
    if days != 0:
        if item.created_utc > rmvafter:
            item.mod.remove()
            continue
    if rmvban:
        if item.author != None:
            if item.author.name != ”[deleted]”:
                if any(reddit.subreddit(home).banned(item.author.name)):
```

```
                        item.mod.remove()
                        continue
    if any(domain in item.url for domain in dblacklist):
        item.mod.remove()
        continue
    if any(word in item.title for word in wblacklist):
        item.mod.remove()
        continue
    if item.is_self:
        if any(word in item.selftext for word in wblacklist):
            item.mod.remove()
```

Listing 4: Implementation of Subreddit Trends

```
# this listing demonstrates the generation of chart for submission scores
count = {}
for item in reddit.subreddit(home).new():
    try:
        date = datetime.datetime.fromtimestamp(item.created_utc).date()
        key = str(date.day) + '-' + str(date.month)
        if key in count:
            count[key][0] = max(count[key][0], item.score)
            count[key][1] = min(count[key][1], item.score)
            count[key][2] += item.score
            count[key][3] += 1
        else:
            if len(count) == days:
                break
            count[key] = [item.score, item.score, item.score, 1]
    except:
        continue

vals = np.array(list(count.values()))
fig = Figure(figsize = (5, 4), dpi = 100)
ax = fig.add_subplot()
ax.plot(list(count.keys()), vals[:, 0], label='max_score')
ax.plot(list(count.keys()), vals[:, 1], label='min_score')
ax.plot(list(count.keys()), vals[:, 2] / vals[:, 3], label='avg_score')
ax.legend()
canvas = FigureCanvas(fig)
```

4 Deployment

Our application was tested for r/kolkata. 450 comments and 50 recent submissions were scanned to obtain a list of 300 users. 1351 overlapping subreddits were found by scanning the individual user history. The highest overlapping subreddits have been listed in Figure 5.

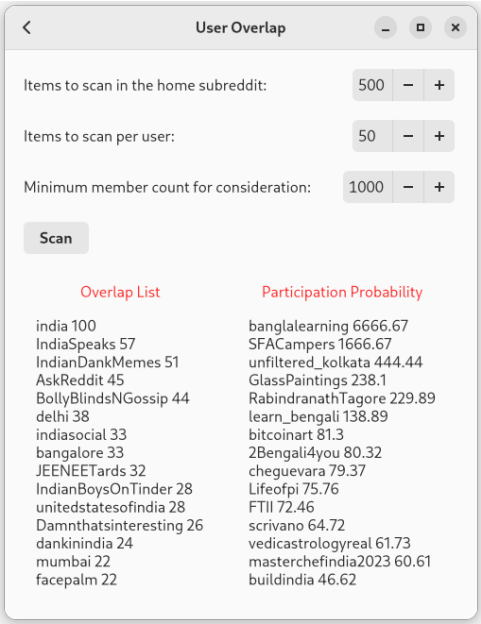

Figure 5: User Overlap Page

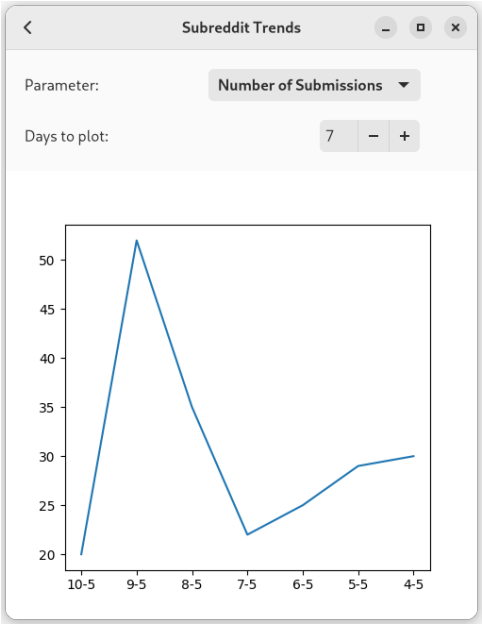

Figure 6: Number of Submissions Chart

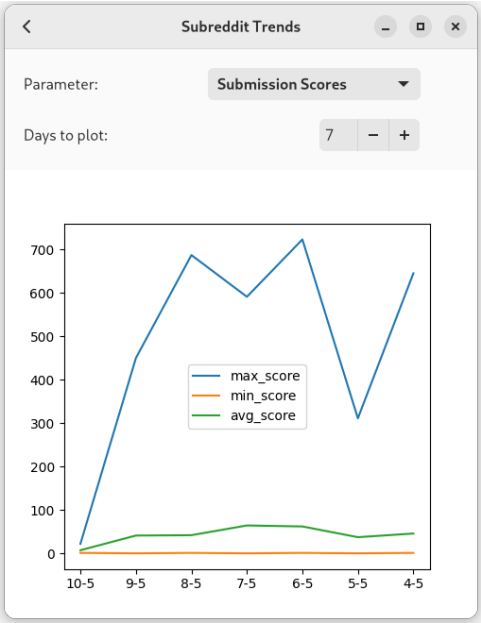

Figure 7: Submission Scores Chart

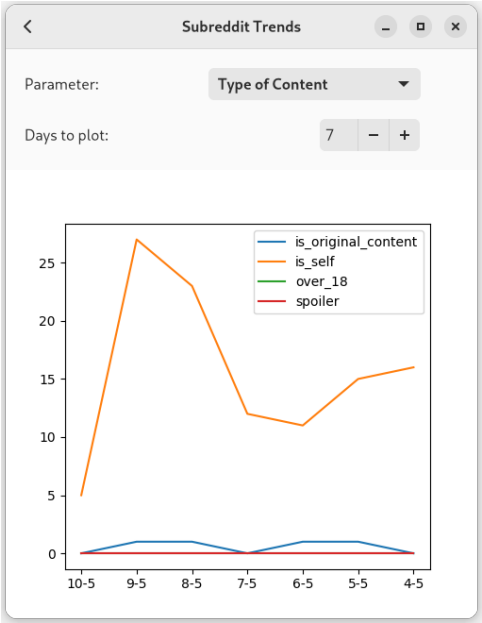

Figure 8: Content Type Chart

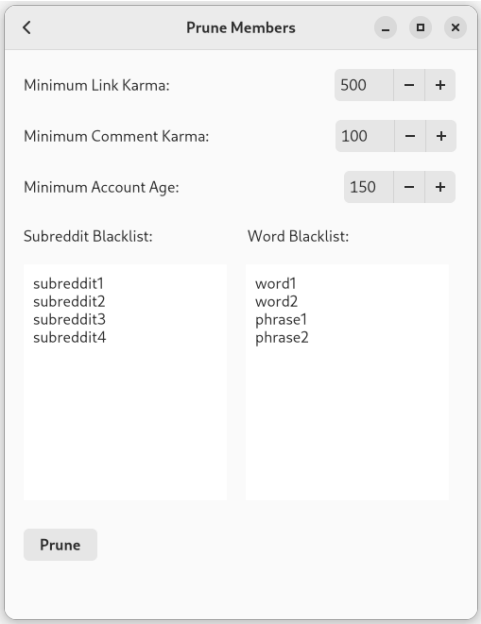

Figure 9: Prune Members Page

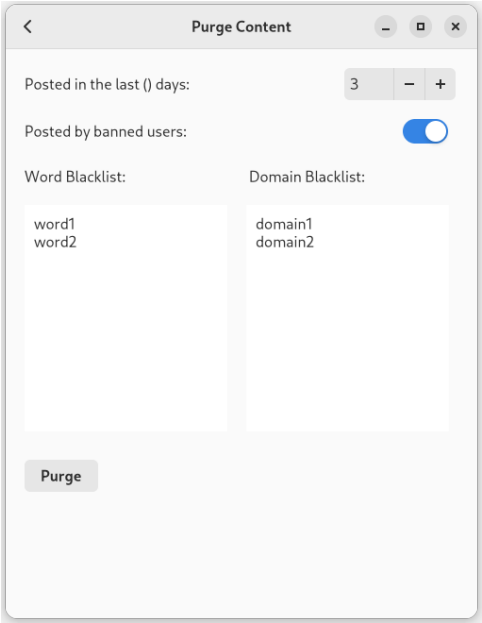

Figure 10: Purge Content Page

5 Conclusion

In this report, we discussed an easy-to-use application for preventing negative interactions. It is worth mentioning that users may suffer from frequent request limit hits by Reddit API, if their account is relatively new. The possibility of classifying items and user history based on the data available in the moderation logs may be explored to further ease the job. Tracking the changes in upvote ratio over time is a key measure in detecting malpractices. However, its implementation would be very costly and is hence beyond the scope of this project.

References

[1] Bryce Boe. Praw: The python reddit api wrapper. Retrieved from https://praw.readthedocs.io/en/latest/, 2022-11-19.

[2] Mark Carman, Mark Koerber, Jiuyong Li, Kim-Kwang Raymond Choo, and Helen Ashman. Manipulating visibility of political and apolitical threads on reddit via score boosting. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 184–190, 2018.

[3] Srayan Datta and Eytan Adar. Extracting inter-community conflicts in reddit. *CoRR*, abs/1808.04405, 2018.

[4] Metrics for Reddit. New subreddits by month - reddit history. Retrieved from https://frontpagemetrics.com/history/month, 2022-05-25.

[5] Reddit Help. Automoderator. Retrieved from https://mods.reddithelp.com/hc/en-us/articles/360002561632-AutoModerator, 2022-11-19.

[6] Reddit Help. Crowd control. Retrieved from https://mods.reddithelp.com/hc/en-us/articles/360038129231-What-is-Crowd-Control, 2022-11-19.

[7] Reddit Inc. Reddit content policy. Retrieved from https://www.redditinc.com/policies/content-policy, 2022-11-18.

[8] Reddit Inc. Reddit user agreement. Retrieved from https://www.redditinc.com/policies/user-agreement, 2022-11-18.

[9] J. Nathan Matias. The civic labor of volunteer moderators online. *Social Media + Society*, 5(2):2056305119836778, 2019.

[10] Alexey Medvedev, Renaud Lambiotte, and Jean-Charles Delvenne. *The Anatomy of Reddit: An Overview of Academic Research*, pages 183–204. 05 2019.

[11] r/Flair_Helper. Flair_helper is currently not accepting new subreddits automatically. Retrieved from https://www.reddit.com/r/Flair_Helper/comments/ro0uqk/flair_helper_is_currently_not_accepting_new/, 2022-10-28.

[12] r/help. Frequently asked questions. Retrieved from https://www.reddit.com/r/help/wiki/faq/, 2022-11-18.

[13] r/SubredditDrama. Rare skirmish breaks out in r/askhistorians. Retrieved from https://www.reddit.com/r/SubredditDrama/comments/rd3zr6/, 2022-11-19.

[14] r/SubredditDrama. r/nonewnormal has been banned. discuss this dramatic happening here! Retrieved from https://www.reddit.com/r/SubredditDrama/comments/pfz0d2/, 2022-11-19.

[15] Purism SPC. Adw – 1. Retrieved from https://gnome.pages.gitlab.gnome.org/libadwaita/doc/main/index.html, 2022-11-19.

[16] GTK Development Team. Gtk – 4.0. Retrieved from https://docs.gtk.org/gtk4/, 2022-11-19.

[17] Ava Trogus, Daniel Fuchs, Aaron Burtle, and Josh Katz. Community interactions on reddit over time. Retrieved from https://courses.cs.washington.edu/courses/cse481ds/21au/resources/trogusava_3717447_71314956_481DS__Reddit_final_report.pdf.

[18] Emily van der Nagel. Faceless bodies: Negotiating technological and cultural codes on reddit gonewild. *Scan: Journal of Media Arts Culture*, 10, 01 2013.