An Exploratory Study of Bot Commits

Tapajit Dey
The University of Tennessee
Knoxville, TN, USA
tdey2@vols.utk.edu

Bogdan Vasilescu
Carnegie Mellon University
Pittsburgh, PA, USA
vasilescu@cmu.edu

Audris Mockus
The University of Tennessee
Knoxville, TN, USA
audris@mockus.org

ABSTRACT

Background: Bots help automate many of the tasks performed by software developers and are widely used to commit code in various social coding platforms. At present, it is not clear what types of activities these bots perform and understanding it may help design better bots, and find application areas which might benefit from bot adoption. Aim: We aim to categorize the Bot Commits by the type of change (files added, deleted, or modified), find the more commonly changed file types, and identify the groups of file types that tend to get updated together. Method: 12,326,137 commits made by 461 popular bots (that made at least 1000 commits) were examined to identify the frequency and the type of files added/ deleted/ modified by the commits, and association rule mining was used to identify the types of files modified together. Result: Majority of the bot commits modify an existing file, a few of them add new files, while deletion of a file is very rare. Commits involving more than one type of operation are even rarer. Files containing data, configuration, and documentation are most frequently updated, while HTML is the most common type in terms of the number of files added, deleted, and modified. Files of the type "Markdown","Ignore List", "YAML", "JSON" were the types that are updated together with other types of files most frequently. Conclusion: We observe that majority of bot commits involve single file modifications, and bots primarily work with data, configuration, and documentation files. A better understanding if this is a limitation of the bots and, if overcome, would lead to different kinds of bots remains an open question.

KEYWORDS

Bots, Bot Commits, Code Commits, Association Rules

ACM Reference Format:
Tapajit Dey, Bogdan Vasilescu, and Audris Mockus. 2020. An Exploratory Study of Bot Commits. In *Proceedings of International Workshop on Bots in Software Engineering (BotSE'20).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Collaborative software development and social coding platforms have seen a surge in bot adoption in recent years [19]. A quick look at the users who create the most number of commits, issues, and/or pull requests reveals that most of them are, in fact, bots [9]. A number of past studies have attempted to categorize the bots that are active in social coding platforms by their origin, purpose, and function (e.g. [10, 16]), however, the activities of the bots have not been directly studied in existing literature.

A study of the commits made by bots can give us valuable insights into their function and application domain, which can be useful for developers planning to adopt bots in their projects in terms of identifying the types of tasks typically handled by bots and the languages they are active in. The developers who design bots might also benefit from this knowledge by identifying the gaps in the current bot landscape in software engineering.

Therefore, in this paper we report on an exploratory study of the commits created by bots in a large dataset [8] of Git open-source projects, introduced by Dey et al. [9]. Specifically, we investigated 12,326,137 commits from 461 popular bots to examine the frequency and the type of files changed by the commits, categorized the commits in terms of the files added, deleted, or modified therein, and identified which types of files frequently get changed together with other types of files.

Our results show that an overwhelming majority (86%) of the bot commits involve only file modification, with 68% of the commits changing only a single file, and only 5% of the commits changed more than 10 files. Data, configuration, and documentation files are the most frequently updated ones, but HTML takes the crown in terms of the number of files added, deleted, and modified. A lot of Java files were found to be added by bots, but not modified frequently, so it most likely was for the purpose of archiving. We also noticed that files of types "Markdown", "YAML", "Ignore List", "JSON", and "Text" are the ones most frequently updated with other types of files in the bot commits that do change more than one file.

We have also compiled a dataset containing information about 150,633,947 file updates by the 12,326,137 bot commits, specifically information about the "blobs"[1] associated with the files before the commit, and the updated blobs. The dataset is available at: https://zenodo.org/record/3699665.

In summary, we find that:

- Bots are mostly taking care of file updates, and update a small number of files per commit, so designing bots with limited scope seems to be the current trend in Software Engineering.
- The majority of bot commits comprises frequent updates to configuration, documentation, and data. Developers planning to adopt bots for their projects might want to consider using bots for similar tasks.
- Bots seem to be more active in Web-interface-related projects, since the majority of bot commits involves changes to HTML, JavaScript, and JSON files. Future researchers might want to investigate the reason for the popularity of bots in this area, which might lead to better design of future bots.

[1] https://git-scm.com/book/en/v2/Git-Internals-Git-Objects

BotSE'20, May 2020, South Korea

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 RELATED WORK

Bots are regularly used in a number of areas like education [2, 15], e-commerce [18], customer service [14], and social media platforms [1]. In Software Engineering, bots are typically used to automate a number of, often tedious and repetitive, tasks performed by software developers and teams. Wessel et al. [19] found 26% of the 351 GitHub projects they studied use bots.

In terms of bot characterization, Lebeuf [16] proposed characterizing the bots by analyzing 22 facets grouped into 3 dimensions: Environmental, Intrinsic, and Interaction. Erlenhov et. al. [10] focused on the 11 well-known bots that support software development, and proposed a taxonomy comprising 4 facets: Purpose (general vs. specialized), Initiation (triggered by users or system or both), Communication (how the bot communicates with other users), and Intelligence (adaptive or static).

A novel method for detecting which commit authors are bots was proposed in Dey et al. [9], which looked at the author name, the commit message, and the files and projects associated with a commit. They compiled a dataset [8] containing information about 13,762,430 commits made by 461 popular bots, each of whom made more than 1000 commits. In this paper, we used that dataset to study the individual commits in further detail. While other studies (e.g. [3]) investigated the activity of human developers, to our knowledge, no study has investigated the bot commits in details.

3 METHODOLOGY

In this section, we describe the data source and the analysis techniques used in this paper.

3.1 Data Source

As mentioned earlier, we used the dataset [8] to obtain a list of the bot commits. The dataset contained information about 13,762,430 commits, however, information about the exact file modification (i.e. the blobs associated with files that were updated by the commit) was not available for a few of them, so, after filtering them out, we were left with 12,326,137 commits.

Detailed information about the files updated by the bot commits were extracted using the World of Code [17] dataset, which is a prototype of an updatable and expandable infrastructure to support research and tools that rely on version control data from the entirety of open source projects that use Git. We used version Q of the dataset for the analysis described in this paper. This data (version Q) was collected on Nov 9 based on updates/new repositories identified on Oct 15, 2019. For further details, please check the WoC website. [2]

3.2 Analysis Method

The data in WoC is stored in the form of mappings between various git objects, and, using the APIs provided by WoC, we constructed a mapping (which we identify as *c2fbb* maps) between a commit, the file(s) modified by that commit, and the old blob associated with each file before the commit, and the new blob associated with the updated version of that file. If either the old blob or the new blob associated with a file was found missing, it meant the file was added or deleted, respectively, by that commit, and if both were found present, it meant that file was modified by that commit.

We started our analysis by investigating the file types that were added, deleted, or modified by each commit. We extracted the file extensions from those files, and used the *linguist* [3] tool to obtain an estimated language classification based on a common open-source model.

To find out what types of files are updated together, we used association rule mining on the file types updated in each commit using the "arules" [13] package in R. Since most of the bot commits were found to update only one type of file, we applied this technique only on the commits that update two or more types of files, so that the result doesn't get overwhelmed by singletons. We used a minimum support of 0.1% and a confidence of 80%, along with a maximum length of 10 types of files for a single association rule for the `apriori` function call. We used the "arulesViz" [12] package in R for visualizing the results of this analysis.

4 RESULT

4.1 Shared Dataset:

We compiled a dataset with the information about the blob updates related to each file updated by one of the bot commits, so that other researchers may also study the bot commits. Details about the structure of the dataset, and how to use the GitHub API to extract the contents of each blob is mentioned in the description of the dataset available through the link in Section 1.

4.2 Categorizing Bot Commits

By observing the *c2fbb* maps, we identified all the files that were updated by the bot commits under consideration, along with the old and new blobs associated with the file, and could also identify which files were added, deleted, or modified by each commit (see Section 3). We categorized the bot commits into 7 categories by the type of file change (addition, deletion, and modification):

(1) Type **A**: Commits involving only file addition.
(2) Type **D**: Commits involving only file deletion.
(3) Type **M**: Commits involving only file modification.
(4) Type **AM**: Commits involving file addition and modification.
(5) Type **DM**: Commits involving file deletion and modification.
(6) Type **AD**: Commits involving file addition and deletion.
(7) Type **ADM**: Commits involving file addition, deletion, and modification.

The relative prevalence of each type of commit is shown in Figure 1, which shows majority (85.98%) of the commits involve only file modifications (Type **M**), while more complex commits (ones with more than one type of file change: Types **AM, DM, AD, ADM**) are relatively uncommon. Interestingly, commits involving only file deletions (Type **D**) are rarer than commits where some files were added and some were modified (Type **AM**), and complex commits involving file deletions (Types **DM, AD, ADM**) are even rarer. Bots that modify files include bots like *GreenKeeper* that update dependency versions for NPM packages, archival bots like *AUR Archive Bot* are responsible for adding a lot of files, while a number file deletions was performed by bots responsible for checking in human commits after running tests, e.g. instances of *Travis CI bots*.

We also looked at the commit sizes for the bot commits, i.e. how many files were changed by each commit. Looking at the overall

[2] http://worldofcode.org

[3] https://github.com/github/linguist

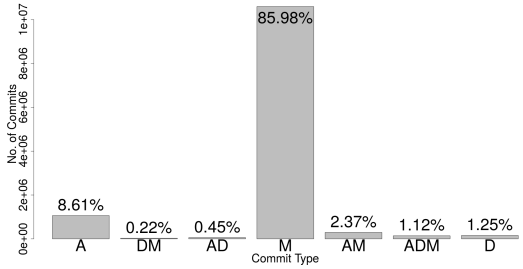

Figure 1: Distribution of the number of commits of each type, with percentage of each commit type shown on the X-axis.

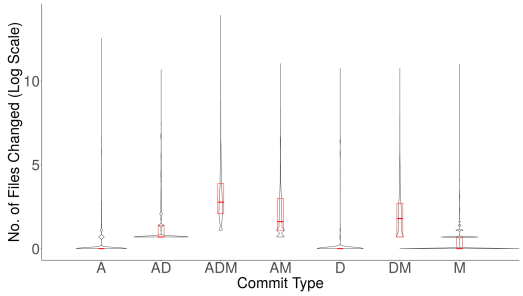

Figure 2: Distribution of the number of files modified by each type of bot commit.

picture, we observed most commits to be of very small size, with the median no. of files changed being 1, but the maximum number of files changed by a commit was observed to be a whooping 1,113,522. We also investigated the distribution of the number of files changed by commits of different types, which is shown in Figure 2 using a violin plot, with the median and 1st and 3rd quantile values shown in a (red) crossbar plot. We observed a highly skewed distribution of the number of files changed for all commit types, however, the more complex commit types were observed to have relatively larger median number of files changed.

While Figure 2 shows that most commits change very few files, we were curious about how many different file types are changed by the commits to get a sense of the degree of heterogeneity in the commits. We found that 10,270,857 (83%) commits have changed only one type of file, while the maximum number of different file types changed by a bot commit was found to be 121.

We can infer from these results that bots are primarily designed for simpler updates involving modification to a few files, which leads us to believe that they have a limited scope, so developers interested in adopting bots might want to consider a similar focus for their application.

4.3 Types of Files changed by Bot Commits

We observed that out of the 150,633,947 instances when a file was changed by one of the bot commits, 50.1%, 40.9%, and 9% are of file modification, file addition, and file deletion respectively.

After inferring the language of the files using the GitHub Linguist tool, [4] we decided to observe what types of files are added, deleted, and modified most frequently by bot commits. Any file type that was not defined in the Linguist tool (e.g. files with extensions

[4]https://github.com/github/linguist

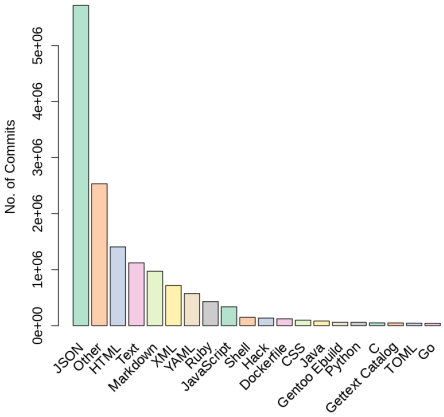

Figure 3: Different file types updated by bot commits.

like `.icloud`, `.AURINFO` etc.) were categorized as "Other", which was primarily comprised of files specific to a particular software. Figure 3 shows the top 20 file types that are updated by different bot commits, with the number of commits that updated a particular file type in the Y-axis. We see that JSON files are updated by the most number of commits, followed by "Other" and HTML files. Overall, we see configuration, documentation, and other data related files are updated more frequently than code related files. Ruby, JavaScript, and Shell are the top three languages updated most frequently. Different types of commits showed very similar distributions to the overall situation for the file types in terms of the number of commits that updated it, so we do not show those plots separately.

We also looked at the number of files of different types that are updated by the bot commits, and we observed that HTML was the most frequently deleted, added, and modified file type, as shown in Figure 4. Java files are among the most frequently added file type, Go files are among the most frequently deleted ones, and JSON files are the second most frequently modified file type. Comparing what we see from Figure 3, we can infer that the average number of HTML files updated per commit is much higher than the average number of JSON files updated per commit, since Figure 3 shows the number of commits that updated a particular file type, whereas Figures 4 show the total number of files (of each type) updated. It also seems from our results that most of the bot updates involve updates to Web interface related files, so bot designers might want to look into expanding automation to other domains as well.

4.4 File Types frequently updated with others

As mentioned earlier, only 17% of all the bot commits change more than one type of file, so investigating what types of files are updated together with other types can give us some insight about these relatively infrequent heterogeneous commits. We applied association rule mining on these commits to address this question, as mentioned in Section 3, and obtained a set of 323 non-redundant association rules, with the minimum, maximum, and median sizes of the rule lengths being 2,6, and 4 respectively. The support for the rules varied between 0.001 and 0.006, with a confidence range between 0.802 and 0.998. The lift for the rules was observed to be between 1.78 and 378.93, with a median of 9.79.

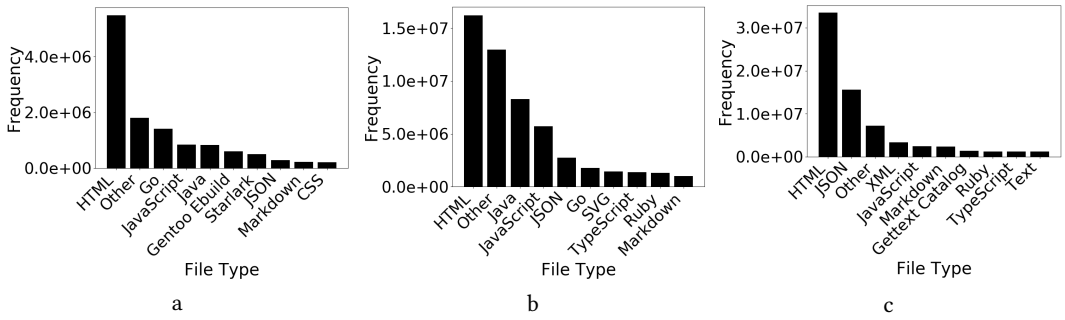

Figure 4: File Types changed by bots: (a): Top 10 deleted, (b)Top 10 added, (c):Top 10 modified.

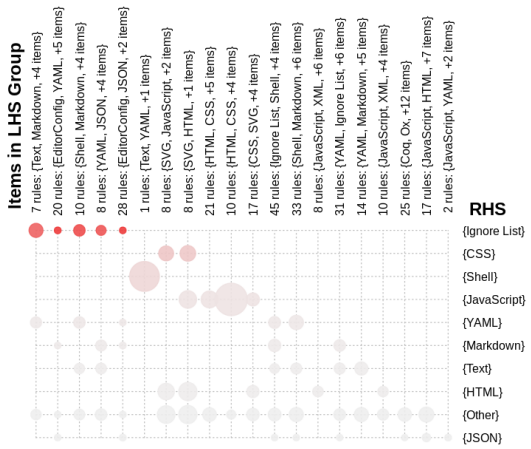

Figure 5: Visualizing grouped association rules

We show a grouped representation of all the rules in Figure 5, where similar rules are grouped together for ease of representation. The size of each circle in the figure corresponds to the support associated with that rule, and the intensity of the color red corresponds to the lift associated with the rule. An interesting observation from this figure is that rules with files of type "Ignore List" (e.g. `.gitignore`) have very high values of lift (these are also the rules with most confidence), i.e. updates to "Ignore List" file types occur more frequently in the commits that update other file types (as shown in Figure 5) as well. The file types on **RHS** column in the figure are the ones that appear more frequently with other file types than by themselves (since the lift for all the rules we have is positive). We also see that files of type "Other" were associated with most of the rules, and rules with "Shell" and "JavaScript" file types in **RHS** have the highest support. We observe that most file types in the rules are related to data, configuration, documentation, and web-design, similar to what we observe overall, and file types that are updated together tend to be complimentary (e.g. HTML with JavaScript and CSS).

We show the top 5 association rules with highest lift values in Table 1. For all 5 rules, "Ignore List" file type was in the **RHS**, and "EditorConfig" was one of the types in **LHS**. The confidence for these rules was found to be very high (0.99), and the associated support values were 0.001 for all of them.

5 LIMITATIONS

We inferred the types of files directly from their names, instead of looking into the contents of the files, which comes with an obvious risk of error. In addition, we faced some challenges while using the Linguist tool for file classification, since some file extensions were found to be linked with multiple file types (e.g. ".gs" files are associated with GLSL, Genie, Gosu, and JavaScript file types). We addressed this problem by adding an entry to all possible types when we encountered such cases, which introduces some error in our result. However, this scenario is not very common (less than 1%), so it should not cause too much of a problem.

We also only focused on the commits by 461 bots that made over 1000 commits each, out of possibly thousands of bots that make code commits, so our results may not generalize for the overall bot population.

Table 1: Top 5 Association Rules by lift

LHS		RHS	support	confidence	lift
{EditorConfig, Markdown, Other, YAML}	=>	{Ignore List}	0.001	0.99	378.93
{EditorConfig, JSON, Other, YAML}	=>	{Ignore List}	0.001	0.99	378.74
{EditorConfig, Other, YAML}	=>	{Ignore List}	0.001	0.99	378.26
{EditorConfig, JSON, Markdown, YAML}	=>	{Ignore List}	0.001	0.99	378.26
{EditorConfig, JSON, Markdown, Other}	=>	{Ignore List}	0.001	0.99	378.09

6 FUTURE WORK

In future, we would like to extend our study of bot commits by looking into the text diffs between the old and the updated blobs associated with each file changed by each commit, which would give us further insight into what exact changes are made by the bots in their commits. Furthermore, we would like to address the issue of multiple IDs related to a bot, by using the methodology proposed in Fry et al. [11], and also investigate how the presence of bots affect the popularity Dey and Mockus [4] of a software and, in turn, affect the number of issues Dey and Mockus [5, 6] and pull request acceptance Dey and Mockus [7].

7 CONCLUSION

In this paper, we investigate the commits created by bots, categorize the commits based on the type of file operation they perform and what types of files they change to understand the types of works performed by bots at present in the context of software engineering, and insights from this study might be valuable in understanding the strengths and limitations of the bots presently active, and can, in future, lead to better design and wider adoption of bots.

REFERENCES

[1] Norah Abokhodair, Daisy Yoo, and David W McDonald. 2015. Dissecting a social botnet: Growth, content and influence in Twitter. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 839–851.

[2] Luciana Benotti, María Cecilia Martínez, and Fernando Schapachnik. 2014. Engaging high school students using chatbots. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*. ACM, 63–68.

[3] Tapajit Dey, Yuxing Ma, and Audris Mockus. 2019. Patterns of effort contribution and demand and user classification based on participation patterns in npm ecosystem. In *Proceedings of the Fifteenth International Conference on Predictive Models and Data Analytics in Software Engineering*. ACM, 36–45.

[4] Tapajit Dey and Audris Mockus. 2018. Are software dependency supply chain metrics useful in predicting change of popularity of npm packages?. In *Proceedings of the 14th International Conference on Predictive Models and Data Analytics in Software Engineering*. ACM, 66–69.

[5] Tapajit Dey and Audris Mockus. 2018. Modeling relationship between post-release faults and usage in mobile software. In *Proceedings of the 14th International Conference on Predictive Models and Data Analytics in Software Engineering*. ACM, 56–65.

[6] Tapajit Dey and Audris Mockus. 2020. Deriving a usage-independent software quality metric. *Empirical Software Engineering* 25, 2 (2020), 1596–1641.

[7] Tapajit Dey and Audris Mockus. 2020. Which Pull Requests Get Accepted and Why? A study of popular NPM Packages. *arXiv preprint arXiv:2003.01153* (2020).

[8] Tapajit Dey, Sara Mousavi, Eduardo Ponce, Tanner Fry, Bogdan Vasilescu, Anna Filippova, and Audris Mockus. 2020. A dataset of Bot Commits. (Jan. 2020). https://doi.org/10.5281/zenodo.3694401

[9] Tapajit Dey, Sara Mousavi, Eduardo Ponce, Tanner Fry, Bogdan Vasilescu, Anna Filippova, and Audris Mockus. 2020. Detecting and Characterizing Bots that Commit Code. *arXiv preprint arXiv:2003.03172* (2020).

[10] Linda Erlenhov, Francisco Gomes de Oliveira Neto, Riccardo Scandariato, and Philipp Leitner. 2019. Current and future bots in software development. In *Proceedings of the 1st International Workshop on Bots in Software Engineering*. IEEE Press, 7–11.

[11] Tanner Fry, Tapajit Dey, Andrey Karnauch, and Audris Mockus. 2020. A Dataset and an Approach for Identity Resolution of 38 Million Author IDs extracted from 2B Git Commits. *arXiv preprint arXiv:2003.08349* (2020).

[12] Michael Hahsler. 2017. arulesViz: Interactive Visualization of Association Rules with R. *R Journal* 9, 2 (December 2017), 163–175. https://journal.r-project.org/archive/2017/RJ-2017-047/RJ-2017-047.pdf

[13] Michael Hahsler, Bettina Gruen, and Kurt Hornik. 2005. arules – A Computational Environment for Mining Association Rules and Frequent Item Sets. *Journal of Statistical Software* 14, 15 (October 2005), 1–25. http://dx.doi.org/10.18637/jss.v014.i15

[14] Mohit Jain, Ramachandra Kota, Pratyush Kumar, and Shwetak N Patel. 2018. Convey: Exploring the use of a context view for chatbots. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 468.

[15] Alice Kerry, Richard Ellis, and Susan Bull. 2008. Conversational agents in E-Learning. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, 169–182.

[16] Carlene R Lebeuf. 2018. *A taxonomy of software bots: towards a deeper understanding of software bot characteristics*. Ph.D. Dissertation.

[17] Yuxing Ma, Chris Bogart, Sadika Amreen, Russell Zaretzki, and Audris Mockus. 2019. World of Code: An Infrastructure for Mining the Universe of Open Source VCS Data. In *IEEE Working Conference on Mining Software Repositories*. papers/WoC.pdf

[18] NT Thomas. 2016. An e-business chatbot using AIML and LSA. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2740–2742.

[19] Mairieli Wessel, Bruno Mendes de Souza, Igor Steinmacher, Igor S Wiese, Ivanilton Polato, Ana Paula Chaves, and Marco A Gerosa. 2018. The power of bots: Characterizing and understanding bots in oss projects. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 182.