World of Code Data Challenge

Audris Mockus
EECS
University of Tennessee
audris@utk.edu

James Herbseb
School of Software Engineering
Carnegie Mellon University
jdh@cs.cmu.edu

Alexander Nolte
Institute of Computer Science
University of Tartu
alexander.nolte@ut.ee

***Index Terms*—Mining Software Repositories, Hackathon**

a) Title: World of Code (WoC) dataset version R.

b) Content: WoC/R represents 123M git repositories from GitHub, GitLab, BitBucket, etc. based on updates/new repositories identified on Mar 7, 2020, and retrieved by Mar 28. It has 2B git commits (42M authors), 8B blobs, and 8.3B trees. The content of these objects is augmented via cross-referencing (a graph) for immediate access to all object references associated with any specific object. For example, all commits that have created a specific blob, all repositories where a specific blob or a specific commit resides in, all commits for a specified author ID, the child commits of a specific commit, and other maps that are impossible to compute without complete data.

Developers often has multiple commit author IDs. WoC solves this problem via a map that associates each author ID with imputed developer identity [1]. Some of these are actually bots [2].

With many repositories being clones or forks, WoC for each repo also provides an estimated project identity for each repository [3]. This results in 69M independent projects.

Time-based selection of commits and selection of projects and authors based on the characteristics of files and activities associated with them is also provided.

WoC pre-calculates basic summaries for each blob by running ctags and stores the results as well as diffs for all 2B commits. Finally, WoC provides blob to import/include/use statements parsed from blobs in 17 programming languages.

c) Size.: The raw data and cross-referencing maps for WoC version R take approximately 200TB on disk.

d) Access.: Due to its large size, WoC provides ways to easily access, sample, and analyze the data based on all the cross-referencing capabilities. Shell, Python, and Perl API can be use on WoC servers via ssh (high performance) and REST APIs for web access. Users need to fill in a short web form for ssh access (see Appendix II).

e) Representativeness.: The objective of WoC is to collect data from all **public git** repositories. WoC has 10% more commits than Software Heritage [4] which aims "to collect, preserve, and share all software that is publicly available in source code form"

f) Specialized tools.: The various APIs to access the data are described in the tutorial [5].

g) Participant needs.: The participants comfortable using terminal would benefit from high performance of ssh access where two simple shell commands showCnt and getValues can be composed with regular shell commands into an arbitrarily complex analysis as described in the tutorial. Participants who disdain terminal can use Python in Jupyter notebooks on WoC servers [6] or run Jupyter notebooks on participants' computers and access WoC via REST APIs [7]. The primary focus of WoC is to provide cross-referencing and completeness of the coverage of the git objects from public repositories. Participants will be responsible for integration of WoC data with other types of data they may need: e.g., issue metadata from GHtorrent, code parsing, statistical modeling, and and other types of data and analysis (See Appendix IV for an example workflow).

h) The research questions.: The purpose of WoC is to enable answering questions that can not be answered without the completeness and cross-referencing of public git data. Research on networks of developers, code, and API's would be benefit the most. For example, questions that require the measurement of activity of the entire OSS at a particular slice of time, identifying all commits/repositories/blobs/developers associated with a specific developer/blob/API, finding all child commits of any commit, or finding all forks/clones of a repository. Simple tasks like finding all commits/blobs associated with a repository are supported but do not require WoC, as "git clone" would suffice. WoC can be used for representative sampling of the developers and projects by developer and project network properties [8]. Such complete picture allows to go past MSR analyses limited by data filtered on the set of non-representaive projects and, therefore, unable to capturing the full scope of developer activity, code reuse, or API usage. Mining questions might involve improving quality of WoC data, its performance, ways to link to other datasets. WoC use examples are in [9] and in publications in Appendix III. The evolution of open source software, developer and project ecosystems, past and current trends, counting and summarizing developers, projects, the source code, and the dynamics of developer and code movements across entire OSS ecosystem are examples of research questions that could not be answered without WoC. Answers to these questions have major practical implications for developers and industry alike: will a library becomes obsolete? Do we have any open source code?

i) Sample data.: Since sampling large graphs is difficult, we provide a Jupyter notebook [10] that investigates blobs/projects/authors associated with a CVE.

Appendix I: The Process

MSR Data Challenges has not provided active support to participants in the past. We propose several ways to engage participants more actively.

1) We will conduct a set of online tutorials during which we will walk MSR Data Challenge participants through the avilable WoC functionality.
2) We will organize an online hackathon for researchers who would like to team up with others on the research problems for this data challenge. The event will provide activities typical of the in-person hackathon virtually. For example, defining research questions, forming teams, scoping problems. This will be done while also providing advice on the best ways to conduct data processing and improve performance. We will provide support in the form of mentors and it will provide the opportunity to work with world-class researchers on relevant problems and research questions. We have previously organized a WoC hackathon and a number of papers in Appendix III resulted from it.
3) We will provide dedicated (issue tracker) support to answer questions and solve issues for the participants of MSR Data Challenge.

We plan to hold the hackathon within one month of the proposal being accepted to ensure there is plenty of time to work on the submissions. We will recruit mentors from past hackathon participants and everyone who had substantial experience of using WoC. Core mentors will be graduate students who have been working on/with WoC.

Depending on the level of interest, we may organize several hackathons to make each of manegeable size. Two of the proposers have done extensive research on hackathons and will advise on the best approach for any size of the groups. We plan to do three tutorials to ensure reasonable participation times for Americas, Europe, and east Asia.

To ensure fair distribution of support among the participants we will instruct hackathon mentors and issue resolvers to distribute effort among participants equally, for example, treating new issues from participants who have not received prior help with high priority. Mentors will be given guidelines on providing help with WoC problems, such as helping debug WocC access, suggesting approaches that improve performance, suggest relationships or other features in WoC that participants may have overlooked. The guidelines will discouraged from suggesting new ideas or correcting problems outside WoC, such as statistical errors. Mentors will be rotated over teams to address potential mentor bias.

We believe that this approach may attract more participation from both industry and academia and lead to more and more interesting results.

Appendix II: The Access Form

MSR Data Challange Hackathon registration

Any identifying information provided here will remain confidential. Your participation in this event is entirely voluntary.

* Required

"I certify that I will not publish personally identifiable information obtained from personally identifiable author information in WoC or use the this information to contact authors without their consent."

* Yes/No

Background

First name

Last name

Email address *

Affiliation

Gender

Female/Male/Non-binary/Prefer not to say/Other

What is highest level of formal education that you have completed until now?

High school diploma or GED/Some college Associate and/or bachelor's degree/Bachelor's degree/Professional degree/Master's degree Doctorate/Prefer not to say

What is your current occupation? (e.g. data scientist, PhD student, post-doc, faculty)

Access to World of Code

This information is required to add you to the corresponding organizations on github and create your login access to the World of Code systems. Github handle:

Preferred login name for World of Code *

Public ssh key * Please refer to https://help.github.com/en/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent for information on how to create it if you do not have one already.

WDo you have an idea you would like to work on during the MSR Data Challenge World of Code hackathon? If yes, please describe it briefly below.

Appendix III: Research enabled by WoC

Amreen, Sadika and Karnauch, andrey and Mockus, Audris. (2019). Developer Reputation Estimator (DRE). ASE '19: Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering. 1082–1085.

Amreen, S and Mockus, A and Zaretzki, R and Bogart, C and Zhang, Y. (2020). ALFAA: Active Learning Fingerprint based Anti-Aliasing for correcting developer identity errors in version control systems.. Empirical software engineering. 25 1136-1167.

Ma, Yuxing and Bogart, Christopher and Amreen, Sadika and Zaretzki, Russell and Mockus, Audris. (2019). World of code: an infrastructure for mining the universe of open source VCS data. MSR '19: Proceedings of the 16th International Conference on Mining Software Repositories. 143–154.

Dey, Tapajit and Ma, Yuxing and Mockus, Audris. (2019). Patterns of Effort Contribution and Demand and User Classification based on Participation Patterns in NPM Ecosystem. PROMISE'19: Proceedings of the Fifteenth International Conference on Predictive Models and Data Analytics in Software Engineering. 36–45.

Zhang, Y and Zhou, M and Mockus, A and Jin, Z. (2019). Companies' Participation in OSS Development - An Empirical

Study of OpenStack. IEEE transactions on software engineering. 1 - 1.

Dey, T and Mockus, A. (2020). Deriving a usage-independent software quality metric. Empirical software engineering. 25 1596–1641.

Ma, Y and Mockus, A and Zaretzki, R and Bichescu, B and Bradley, R. (2020). A Methodologyfor Analyzing Uptake of Software Technologies Among Developers. IEEE transactions on software engineering.

Mockus, Audris. (2019). Insights from open source software supply chains. ESEC/FSE '19: 27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 3.

Amreen, Sadika and Mockus, Audris. (2017). Experiences on Clustering High-Dimensional Data using pbdR. Proceedings of the 1st International Workshop on Software Engineering for High Performance Computing in Computational and Data-enabled Science Engineering. 9 to 12.

Valiev, Marat and Vasilescu, Bogdan and Herbsleb, James. (2018). Ecosystem-level determinants of sustained activity in open-source projects: a case study of the PyPI ecosystem. Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 644 to 655.

Dey, Tapajit and Mockus, Audris. (2018). Modeling Relationship between Post-Release Faults and Usage in Mobile Software. PROMISE'18 Proceedings of the 14th International Conference on Predictive Models and Data Analytics in Software Engineering. 56 to 65.

Ma, Yuxing and Bogart, Chris and Amreen, Sadika and Zaretzki, Russell and Mockus, Audris. (2019). World of code: an infrastructure for mining the universe of open source VCS data. MSR '19 Proceedings of the 16th International Conference on Mining Software Repositories. 143-154.

Dey, Tapajit and Mockus, Audris. (2018). Are Software Dependency Supply Chain Metrics Useful in Predicting Change of Popularity of NPM Packages?. PROMISE'18 Proceedings of the 14th International Conference on Predictive Models and Data Analytics in Software Engineering. 66 to 69.

Dey, Tapajit and Ma, Yuxing and Mockus, Audris. (2019). Patterns of Effort Contribution and Demand and User Classification based on Participation Patterns in NPM Ecosystem. In Proceedings of the 15th International Conference on Predictive Models and Data Analytics in Software Engineering.

Zhu, Jiaxin and Zhou, Minghui and Mockus, Audris. (2016). Effectiveness of code contribution: from patch-based to pull-request-based tools. Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering. 871 to 882.

Amreen, Sadika and Bichescu, Bogdan and Bradley, Randy and Dey, Tapajit and Ma, Yuxing and Mockus, Audris and Mousavi, Sara and Zaretzki, Russell. (2019). A Methodology for Measuring FLOSS Ecosystems. Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability. 1-29.

Zhou, Minghui and Chen, Qingying and Mockus, Audris and Wu, Fengguang. (2017). On the scalability of Linux kernel maintainers' work. ESEC/FSE 2018. 27 to 37.

Bradley, R., Ma, Y., Bicescu, B., Zaretzky, R., and Mockus, A. Coordinating Interdependencies in an Open Source Software Project: A Replication of Lindberg, et al., AIS Transactions on Replication Research, submitted.

Fry, T., Dey, T., Karnauch, A., Mockus, A. (2020). A Dataset and an Approach for Identity Resolution of 38 Million Author IDs extracted from 2B Git Commits. MSR'2020.

Mockus, A., Spinellis, D., Kotti, Z., Dusing, G. J. (2020). A Complete Set of Related Git Repositories Identified via Community Detection Approaches Based on Shared Commits..

Spinellis, D., Kotti, Z., Mockus, A. (2020). A dataset for github repository deduplication. MSR'2020.

Dey, T., Mockus, A. (2020). Which Pull Requests Get Accepted and Why? A study of popular NPM Packages. Accepted at ESEM'2020.

Dey, T., Mousavi, S., Ponce, E., Fry, T., Vasilescu, B., Filippova, A., Mockus, A. (2020). Detecting and Characterizing Bots that Commit Code. MSR'2020.

Dey, T., Vasilescu, B., Mockus, A. (2020). An Exploratory Study of Bot Commits. MSR'2020.

Zhang, Y., Zhou, M., Mockus, A., Jin, Z. (2020). Companies' Participation in OSS Development-An Empirical Study of OpenStack. ICSE'2020, Journal First.

Appendix IV

The analysis of Dey et al, on models of pull request acceptance in NPM uses WoC to determine the public repositories where PR creators have previously contributed to. Only WoC provides such capability due to the comprehensive collection of open source projects and the cross-reference where for each developers all commits and projects have been identified.

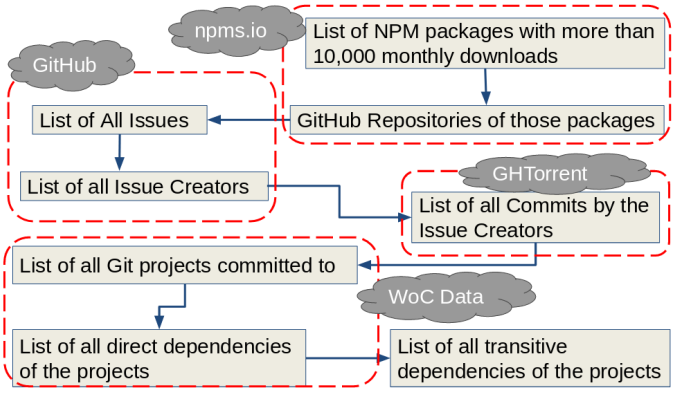

Fig. 1. Example analysis workflow that includes WoC. Thanks to T. Dey

References

[1] T. Fry, T. Dey, A. Karnauch, and A. Mockus, "A dataset and an approach for identity resolution of 38 million author ids extracted from 2b git commits," in *IEEE Working Conference on Mining Software Repositories: Data Showcase*, May 2020. [Online]. Available: https://arxiv.org/abs/2003.08349

[2] T. Dey, S. Mousavi, E. Ponce, T. Fry, B. Vasilescu, A. Filippova, and A. Mockus, "Detecting and characterizing bots that commit code," in *IEEE Working Conference on Mining Software Repositories*, May 2020. [Online]. Available: https://arxiv.org/abs/2003.03172

[3] A. Mockus, D. Spinellis, Z. Kotti, and G. J. Dusing, "A complete set of related git repositories identified via community detection approaches based on shared commits," in *IEEE Working Conference on Mining Software Repositories: Data Showcase*, May 2020. [Online]. Available: https://arxiv.org/abs/2002.02707

[4]

[5] [Online]. Available: https://github.com/woc-hack/tutorial

[6] [Online]. Available: https://github.com/woc-hack/msr-challenge/PYJupyter.ipynb

[7] [Online]. Available: https://github.com/woc-hack/msr-challenge/blob/master/RESTJupyter.ipynb

[8] A. Tutko, A. Henley, and A. Mockus, "More effective software repository mining," in *http://arxiv.org/abs/2008.03439*.

[9] Y. Ma, C. Bogart, S. Amreen, R. Zaretzki, and A. Mockus, "World of code: An infrastructure for mining the universe of open source vcs data," in *IEEE Working Conference on Mining Software Repositories*, May 26 2019. [Online]. Available: papers/WoC.pdf

[10] [Online]. Available: https://github.com/woc-hack/msr-challenge/CVEJupyter.ipynb