2017 IEEE/ACM 39th IEEE International Conference on Software Engineering Companion

Mining Software Engineering Data from GitHub

Georgios Gousios
Department of Software Technology
Delft University of Technology
Delft, The Netherlands
g.gousios@tudelft.nl

Diomidis Spinellis
Department of Management Science and Technology
Athens University of Economics and Business
Athens, Greece
dds@aueb.gr

***Abstract*—GitHub is the largest collaborative source code hosting site built on top of the Git version control system. The availability of a comprehensive API has made GitHub a target for many software engineering and online collaboration research efforts. In our work, we have discovered that a) obtaining data from GitHub is not trivial, b) the data may not be suitable for all types of research, and c) improper use can lead to biased results. In this tutorial, we analyze how data from GitHub can be used for large-scale, quantitative research, while avoiding common pitfalls. We use the GHTorrent dataset, a queryable offline mirror of the GitHub API data, to draw examples from and present pitfall avoidance strategies.**

***Index Terms*—GitHub; GHTorrent; empirical software engineering; Git**

I. Description

GitHub is a collaborative code hosting site built on top of the git version control system. It includes a variety of features that encourage teamwork and continued discussion over the life of a project. GitHub uses a "fork & pull" collaboration model [1], where developers create their own copies of a repository and submit requests when they want the project maintainer to incorporate their changes into the project's main branch, thus providing an environment in which people can easily conduct code reviews. Every repository can optionally use GitHub 's issue tracking system to report and discuss bugs and other concerns. GitHub also contains integrated social features: users are able to subscribe to update by "watching" projects and "following" other users, resulting in a constant stream of data about people and projects of interest. The system supports user profiles that provide a summary of a person's recent activity within the site, such as their commits, the projects they forked or the issues they reported.

Due to the combination of reasons such as data availability, data homogeneity and volume, GitHub has become both the target of choice and the source of data for various research efforts, ranging from distributed collaboration [1] to deep learning on software data [2]. However, GitHub data do not come for free for researchers: initially, GitHub is imposing limits on their API, which, given the volume of interesting projects, can put a significant delay on data acquisition; the technicalities of the retrieval complicate the acquisition process. Moreover, there is no data schema (GitHub is only exposing its data as JSON responses through a REST API), while the data only represent the current project state. In addition, selection and filtering of GitHub data imposes threats to the quality of the study itself; as a result, many GitHub - bound studies suffer from data validity issues [3], [4].

A notable archiving attempt, which emerged through the repository mining community, is the GHTorrent project [5], a scalable, off-line mirror of all data offered through the GitHub API. GHTorrent follows the GitHub event stream and systematically retrieves from it all data, their metadata and their dependencies. It then processes and stores all retrieved items in a relational database, while also storing the original data in a MongoDB database. GHTorrent offers to interested researchers both downloads of the corresponding database dumps (currently, >15 TB of data) and online access facilities (including live database access and Google BigQuery).[1] GHTorrent has been very successful: more than 200 researchers have subscribed and used the online access points, 33% of all empirical research publications on GitHub are based on it [4], while it is being used in production by companies, such as Microsoft.

As GHTorrent is becoming the *de facto* standard dataset for large scale quantitative analysis for GitHub data, we believe that it is crucial for researchers to know how to use GHTorrent to sample for projects, how to treat the data and how to avoid common pitfalls in order to minimize the risk of doing unsound research.

In the tutorial, we address the following topics.

- GitHub data collection strategies, including querying the API, using online services such as GitHub Archive and GHTorrent.
- Using GHTorrent to sample appropriate repositories for various types of research questions.
- Writing, managing, and optimizing complex and expensive relational queries on GHTorrent relational data.
- Using GHTorrent effectively: understanding the data collection challenges and avoiding common pitfalls.
- Copyright and privacy issues when using the GitHub data.

II. Speaker Biographies

The two speakers have pioneered the use of data from GitHub in software engineering research with the introduction of the GHTorrent data collection framework [6]. They have both been active on GitHub related research ever since. Apart from creating GHTorrent, the speakers have studied the pull-request based distributed software development practice [1],

[1] https://bigquery.cloud.google.com/queries/ghtorrent-bq

978-1-5386-1589-8/17 $31.00 © 2017 IEEE
DOI 10.1109/ICSE-C.2017.164

co-created an automated work-prioritization framework for pull request based projects [7], co-examined the build and test practices of projects on TravisCI [8] and co-documented the pitfals of using GitHub-based datasets [3].

Georgios Gousios is an assistant professor at the Web Information Systems group, Delft University of Technology. His research interests include software engineering, software analytics and programming languages. He works in the fields of distributed software development processes, software quality, software testing, developer productivity assessment, research infrastructures and software security. His research has been published in top venues, where he has received four best paper awards and various nominations. In total, he has published more that 50 papers and also co-edited the "Beautiful Architectures" book (OReilly, 2009). He is the main author of the GHTorrent data collection and curration framework and the Alitheia Core repository mining platform. Georgios holds and MSc from the university of Manchester and a PhD from the Athens University of Economics and Business, both in software engineering. In addition to research, he is also active as a speaker, both in research and practitioner oriented conferences.

Diomidis Spinellis is a Professor in the Department of Management Science and Technology at the Athens University of Economics and Business, Greece. His research interests include software engineering, IT security, and cloud systems engineering. He has written two award-winning, widely-translated books: *Code Reading* and *Code Quality: The Open Source Perspective*. His most recent book is *Effective Debugging: 66 Specific Ways to Debug Software and Systems*, which was published as part of Addison-Wesley's *Effective Software Development Series* in 2016. Dr. Spinellis has also published more than 200 technical papers in journals and refereed conference proceedings, which have received more than 5000 citations. He served for a decade as a member of the IEEE Software editorial board, authoring the regular *Tools of the Trade* column. He has contributed code that ships with macOS and BSD Unix and is the developer of CScout, UMLGraph, ckjm, dgsh, and other open-source software packages, libraries, and tools. He holds an MEng in Software Engineering and a PhD in Computer Science, both from Imperial College London. Dr. Spinellis has served as an elected member of the IEEE Computer Society Board of Governors (2013—2015), and is a senior member of the ACM and the IEEE. From January 2015 he is serving as the Editor-in-Chief for IEEE Software.

III. Related presentations

- The GHTorrent dataset and toolsuite[2]
- Mining GitHub for Fun and Profit[3]
- The issue #32 incident[4]
- Working Effectively with Pull Requests[5]

[2]https://speakerdeck.com/gousiosg/the-ghtorrent-dataset-and-toolsuite
[3]https://speakerdeck.com/gousiosg/mining-github-for-fun-and-profit
[4]https://speakerdeck.com/gousiosg/the-number-issue32-incident
[5]https://speakerdeck.com/gousiosg/working-effectively-with-pull-requests

Acknowledgement

The project associated with this work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732223.

References

[1] G. Gousios, M. Pinzger, and A. v. Deursen, "An exploratory study of the pull-based software development model," in *ICSE 2014: Proceedings of the 36th International Conference on Software Engineering*. New York, NY, USA: ACM, 2014, pp. 345–355.

[2] X. Gu, H. Zhang, D. Zhang, and S. Kim, "Deep API learning," in *FSE 2016: Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. New York, NY, USA: ACM, 2016, pp. 631–642.

[3] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. German, and D. Damian, "An in-depth study of the promises and perils of mining GitHub," *Empirical Software Engineering*, vol. 21, no. 5, pp. 2035–2071, 2016.

[4] V. Cosentino, J. Luis, and J. Cabot, "Findings from GitHub: methods, datasets and limitations," in *MSR 2016: Proceedings of the 13th International Workshop on Mining Software Repositories*. ACM, 2016, pp. 137–141.

[5] G. Gousios, "The GHTorrent dataset and tool suite," in *MSR 2013: Proceedings of the 10th Working Conference on Mining Software Repositories*, May 2013, pp. 233–236.

[6] G. Gousios and D. Spinellis, "GHTorrent: GitHub's data from a firehose," in *MSR 2012: Proceedings of the 9th Working Conference on Mining Software Repositories*, M. W. Godfrey and J. Whitehead, Eds. IEEE, Jun. 2012, pp. 12–21.

[7] E. van der Veen, G. Gousios, and A. Zaidman, "Automatically prioritizing pull requests," in *MSR 2015: Proceedings of the 12th Working Conference on Mining Software Repositories*, May 2015, pp. 357–361.

[8] M. Beller, G. Gousios, and A. Zaidman, "TravisTorrent: Synthesizing travis CI and GitHub for full-stack research on continuous integration," in *MSR 2017: Proceedings of the 14th Working Conference on Mining Software Repositories*, 2017.