ReBOC: Recommending Bespoke Open Source Software Projects to Contributors

Denae Ford
Microsoft Research
Redmond, WA, USA
denae@microsoft.com

Nischal Shrestha
NC State University
Raleigh, NC, USA
nshrest@ncsu.edu

Thomas Zimmermann
Microsoft Research
Redmond, WA, USA
tzimmer@microsoft.com

***Abstract*—Open Source Software for Social Good (OSS4SG) projects are projects that address a societal need and target people who need help. These projects often address high-impact humanitarian causes such as curating local health resources during a global pandemic, informing the public on the structural integrity of buildings, and encouraging civic engagement in times of strife. These projects carry a high intrinsic reward for contributing but are hard to find—prior research has shown that one of the the top challenges for contributors is not knowing where to find good projects to work on. Currently, contributors must manually search and assess whether projects align with their growing technical skills and intended impact interests.**

In this paper, we describe a recommendation system that automatically recommends OSS4SG projects for contributors based on their activity and project-related information. To score and rank projects, we calculated scores based on four signals: technical skills, interests, social ties, and recency of project activity. We performed an offline validation of the recommendation system using standard evaluation metrics such as the hit rate ratio. Results show that the signals are effective in producing a ranked list of OSS4SG projects for contributors, with room for improvement. Finally, we conducted a formative study with contributors to better understand their process of project discovery, validate our findings, and identify additional signals for future work to improve recommendations.

***Index Terms*—OSS, open-source, social good, open source for social good, recommender systems**

I. Introduction

OSS has a sustainability problem due to long-time contributors transitioning out of projects [1] and existing contributors lack of interest to stay engaged [2]. Many developers have found themselves at a crossroads where many tools our technology-driven society depend on do not have the developers to keep them not only surviving, but thriving [3]. This is especially the case in high-stakes projects such as Open Source Software for Social Good (OSS4SG), where there are real people depending upon these resources [4], [5]. One such project is Public Lab [1] which is a community and non-profit working on democratizing science and giving citizens the tools they need to monitor their local air quality and disaster response. Another is Reach4Help [2] whose goal is to connect global citizens with local volunteers and help organizations when they need it most. But how do we keep projects like the aforementioned sustainable? Two approaches we have taken are to 1) detect and surface important skills it takes to be a successful contributor [6], [7] and 2) to help these projects retain the talent they already have [8]. Although these are helpful, there is still a key question we have yet to explore: how do we get projects access to the contributors they need to keep growing and evolving? Literature has shown that there are multiple roads to success in OSS [9] and that will to help others is a strong motivator [10]. Our goal is help connect contributors interested in having this type of impact with the opportunities to do so.

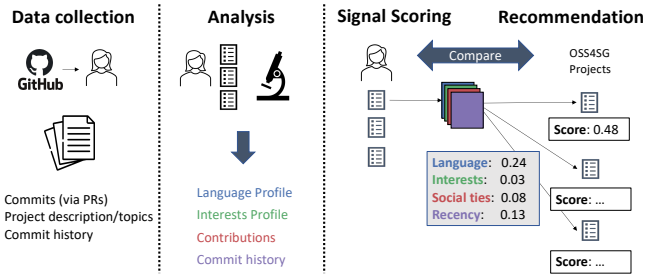


Fig. 1. Recommendation System Overview

To help support this process of finding projects, we built a recommendation system for contributors to identify these projects. Prior literature has indicated that recommendation systems in software can serve as resource to help developers navigate new ways of working [11]. Thus, we built ReBOC, which stands for **Re**commending **B**espoke **O**SS to **C**ontributors. In this tool we use four core signals to make meaningful recommendation to contributors: 1) their previous programming language experience in other projects, 2) topics covered in previous projects they contributed to, 3) their relationship with collaborators, and 4) their recent contribution activity. These signals are strategically based on high-visibility attributes that developers have been inclined to use when navigating the process of deciding to contribute to a new project. Building a tool that can factor in social impact as well as broader OSS project selection factors presents a unique

[1]https://publiclab.org
[2]https://reach4help.org/

978-1-6654-4214-5/22/$31.00 ©2022 IEEE

opportunity to support the general OSS community and those in the niche. To evaluate the usefulness of these signals used in ReBOC, we conducted both an offline validation and formative study with recent OSS4SG contributors. From our offline validation, we find that signals such as the recency of contributions has had the highest hit ratio (in isolation and when combined with other signals) in successful project recommendations. From our formative study, we find that the majority of our participants were satisfied with the recommended projects and provided rich insights into their personal strategies for identifying projects.

This paper makes the following contributions:

- ReBOC, a tool that uses multiple activity-based signals to recommend OSS projects to contributors (Section III).
- An initial evaluation of ReBOC with an offline validation (Section IV) and open source contributors (Section V). ReBOC's hit rate is up to 63.0% for the Top-20 projects.

II. Background

As the concept of OSS grows in popularity so has the challenge of identifying opportunities to contribute. To help mitigate some of the challenges, many organization have tried their hand at amplifying opportunities to newcomers. For instance, tools like GoodFirstIssue.dev [12], Codetribute [13] and FirstTimersOnly.com [14] curate and highlight tasks in popular open source projects to find reasonable first contributions for newcomers. Empirical investigations into the challenges of experts recommending tasks to newcomers include challenges identifying newcomer interests, expertise, and previous contributions [15]. Likewise, Prana et al. propose a project recommendation that considers a variety of community engagement quality, contributor demographic, and social activity metrics [10]. In a similar vein, Qiu et al. take mixed-methods approach to understanding the attributes that contributors look for when identifying these projects finding that a combination of easily quantifiable activity and more challenging to assess factors of openness played a significant role in selection [16]. Overall, these approaches highlight the strategy that goes into identifying the first task and project.

To supplement these individually curated resources, there are also tools that researchers have built to automatically track and recommend projects. For instance, Hu et al. used language-specific popular repositories to track monthly trends of OSS activity [17]. Further, Badashian et al. used a similar interpretation of popular influence to understand how relationships drive what projects people decide to work on [18]. Taking a more behavior-centric approach, Liu et al. identified project features and developer experience to recommend OSS projects [19]. Likewise, Zhang et al. has used developer interaction behaviors such as project membership to provide recommendations [20]. We build on these approaches where we combine project-specific attributes and developer-specific experiences that highlight the goals of a project to recommend projects.

III. ReBOC: The Recommendation System

In the following section, we describe the data collected, signals analyzed, and scoring process that goes into our tool. Figure 1 shows an overview of the recommendation tool.

A. Data Collected

To build our dataset of core projects, we gathered activity from participants from Huang et al.'s dataset on OSS projects [21]. We will call this set *Input-Projects*. We used this dataset of projects as it is one of the most recent datasets of open source for social good projects. From these projects, we then gathered project and individual participant data from the GitHub REST API [22] and GHTorrent [23]. We collected commits of each contributor the project, project descriptions, and topics. Likewise, we gathered the commit history of individual contributors to their other projects as well as the project descriptions and topics of other OSS projects they have worked on.

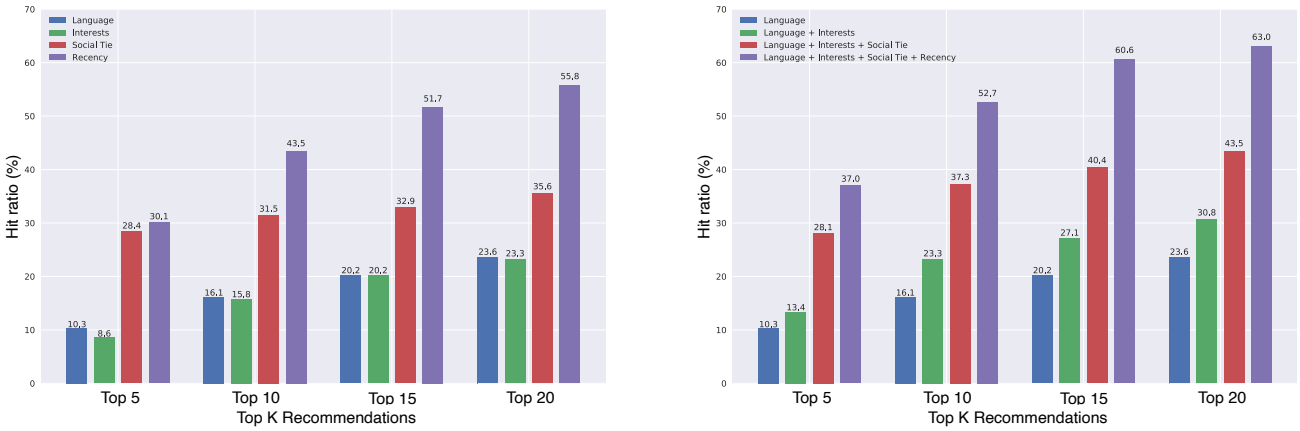
B. Signals Analyzed and Signal Scoring

To build our recommendation system, we then used the collected metadata to measure signals of activity. We intentionally selected high-visibility signals available from a GitHub repository page that a contributor would likely review before selecting a project to work on. We used the following 4 core signals:

a) Language Profile: This signal aims at understanding the technical skills of OSS contributors [6]. For simplicity, we focused on the programming languages that each contributor used. To compute this signal, we used Linguist [24] to classify all the files a contributor has interacted with. Linguist is GitHub's open source tool for highlighting the languages contributors have used in repositories as shown on the GitHub.com profile. To calculate a language score we used the following equation $|CL \cap PL| * PL\_weight$; where CL is contributors languages used, PL is candidate project's language, and PL_weight is the ration of number of PL files to total number of files.

b) Interests Profile: This signal includes the project descriptions and topics. The interest profile included the title of the project and the self-reported description of the project. We calculated the score by using TF-IDF (Term Frequency-Inverse Document Frequency) which is a simple bag-of-words approach where we represent the description/topics as word vectors so that we can calculate similarity between a contributor's corpus and the candidate project's description and topics. To calculate a score for the interest profile, we used the following equation $Avg(\textit{TF-IDF Similarity}(CD, PD))$; where CD is the contributed project description and PD is the candidate's project description.

c) Social Ties: Contributors are more likely to contribute to previous collaborations as observed by Gerosa et al.'s work on what drives contributors [25]. We calculated a social tie score using a similar strategy as Liu et al. [19]. The score was the number of collaboration projects divided by the number of projects someone contributed to. Essentially this score



(a) Isolated Signal Scores

(b) Cumulative Signal Scores

Fig. 2. Top k hit ratio percentage of project recommendations.

measures how often contributors have collaborated with the project owner in the past. A project is ranked higher if a contributor has collaborated more frequently with a package owner in multiple projects.

d) Recency: Contributors may prefer to contribute to active projects with recent activity. To look at recency, we use project commits as a proxy. We used the following equation to calculate a recency score $1/(OT - LCT)$ where *OT* is the time the recommendation is made and *LCT* is the latest commit time of the candidate project. If there is a big gap between the two times, then there is a lower score and the project is less relevant. If there's a small gap, then there is a higher score and the project is more relevant. We use days to measure the time difference. For the offline experiments, we used as recommendation time, either the time of first commit to a candidate project or if not available, the time of the first commit in to a project in *Input-Projects*.

C. Recommendations

The final step of REBOC is to use the information gathered in the signals analysis and scoring phase to make project recommendations. The tool first computes all *individual* signal scores and then aggregates them into a final *cumulative* score. This score is then used to rank the OSS4SG projects. Currently recommendations are presented in a CLI, but can be presented in multiple forms. For this paper, we tested different configurations of REBOC: each signal in isolation and several signal combinations with a cumulative score (see Section IV).

IV. The Offline Validation

To test the effectiveness of the signals used, we conducted an offline validation of the project recommendations. To conduct this evaluation, we split our projects into a training set and a testing set. The training set consisted of the 292 contributors and the projects they contributed to *before contributing* to one of the OSS4SG projects we originally identified them from. The test set includes all 432 OSS4SG projects (gathered by [21] from [5], [26]) we want to recommend based on our analysis. Finally, to mitigate data leakage we inspected both datasets to ensure no overlap of data.

For the offline validation of our tool, we computed recommendations on all contributors and calculated the hit ratio percentage for the top k recommended projects lists (e.g., Top 5, 10, 15, 20). We calculated the hit ratio as the number of users whose target project is listed in the top k list divided by the number of total users. For example, assume that we have two users, zendaya365 and kasey456, and we are looking to see if there is a hit in the Top 5 of projects. We enter their username in the tool to create a list of recommend projects. For zendaya365, the tool recommends a project in the Top 5 that zendaya365 later contributed to, therefore we consider this recommendation a hit. For kasey456, the Top 5 does not contain a project that kasey456 worked on (no hit). This example would result in a 50% hit ratio (1 successful hit out of 2 recommendations made).

In the offline validation, we calculated the hit ratio percentage for each signal in isolation (Figure 2.a) and cumulatively (Figure 2.b) across the top k recommended projects. In both figures across recommendation lists, language profiles alone resulted in the lowest hit ratio. Likewise, the interest profile alone (8.6%) or even when combined with the language profile (13.4%) only marginally improved the rate of success. Generally we see that the hit ratio does improve as we incrementally add signals. Overall, as an isolated signal or in combination with the other signals, the recency signal achieved the highest hit rate getting up to 63% in the Top 20.

V. Formative Study

For an initial evaluation of the REBOC tool, we conducted three semi-structured interviews. We solicited participants from contributors to OSS4SG projects on GitHub. From the 432 OSS4SG projects, we randomly selected contributors for whom we could recommend at least one relevant project with high confidence. We invited 48 contributors to participate in

an interview but only three accepted. In each interview, we asked the participants C1, C2, and C3 how they go about discovering projects (signals), what kinds of pain points they have, and general preferences of what should be part of a personalized project recommendation. We then showed two projects and asked participants if they would contribute (or not) and why. The two projects were personalized for each participant. The first project was predicted to be relevant by ReBOC, the second project was predicted to be not relevant.

A. Results

Participant C1 was driven by learning by doing and wanted to learn by contributing to an open source project. The signals that were important to C1 when selecting projects were the organization and the owner as well as the reputation behind he owner. In addition, C1 valued active projects and was looking at the most recent commit time. Other signals important to C1 were README files and the contribution guide. C1 preferred project recommendations based on relevant languages that they have used. C1 suggested using collaborative filtering for recommendations (*"devs that use package A also use package B"*). C1 mentioned that projects outside of their direct interest can be inspiring, even when they don't necessarily contribute (*"my friend's project made me want to pick up Arduino"*).

Participant C2 was a goal driven contributor. They wanted to find projects that they can use for their own projects, because most of the time they were maintaining their own projects. The signals that C2 mentioned were very similar to the signals mentioned by C1. The main difference was that C2 additionally looked at the comments within the GitHub issues and pull requests. This was to gauge how well the maintainers are handling outside contributions. C2 also thought that filtering based on relevant languages is important. but was also interested in projects in similar languages. C2 was open to project recommendations outside their main work (*"Sometimes, I am interested in projects that aren't about maps like frontend libraries in JavaScript"*). C2 pointed out that toy projects are a pain point when selecting projects and having a way to filter out toy projects would be useful.

Participant C3 was passionate about social good projects and worked on several OSS4SG projects and even started one in their own country. The signals were similar to C1 and C2. This participant also looked at *closed* issues when where they were trying to decide which open issue to work on. They did this because they wanted to see how successful the issues were in getting resolved. This suggests that C3 values the community and organization behind a project. C3 did not want projects recommended to them based on the projects that they've recently worked on and instead was looking for novelty (*"People get bored by the same types of projects"*). C3 also suggested that it would be good to know which community a project belongs to (*"You will find a pattern of what kind of people work on certain projects ... we behave the way how our surrounding behaves"*).

TABLE I
WOULD PARTICIPANTS CONTRIBUTE TO A PROJECT?

	C1	C2	C3
Relevant project (expected: ✓)	?	✓	✓
Irrelevant project (excepted: ×)	×	×	×

B. Takeaways

We showed each participant two projects identified by ReBOC and asked whether they would contribute to the project (see Table I). For the *related* projects, C2 and C3 said that they would contribute (✓). While C1 was interested in the recommended project, C1 was unsure about contributing because the most recent commit in the project was not recent and several months ago. For the *unrelated* projects, C1, C2, and C3 were not interested in contributing (×). Even though this is a small sample size, it suggests that ReBOC can effectively distinguish between relevant and irrelevant projects for a contributor.

The interviews suggested that different personas (learning by doing, goal driven) might affect how contributors search for and select projects. Social and community aspects are increasingly important (C1 and C3). Issues and pull requests were used by several contributors to select projects (C2 and C3). The participants pointed out that the location of recommendations is important. While GitHub does have a page to explore projects, both C1 and C2 were not aware of the page.

VI. Limitations

We acknowledge that there are several limitations to this work. Additional signals or different ways of measuring the signals could have been used. For this paper, we primarily focused on signals that the contributor would see when they first come across a project on GitHub. The dataset for the experiments was relative small with 432 total projects and only 292 contributors who had enough data available. Scaling up this research and evaluating on a larger sample will be an important next step. And finally, the number of interviews was too small to identify general themes and trends. Additional interviews may provide additional insights into how contributors look for projects and lead to either new signals or combinations of existing signals.

VII. Conclusions

Research has shown that finding good projects is a challenge for contributors. To help contributors find projects that align their interests and skill, we describe a recommendation system that automatically recommends OSS4SG projects based on four signals: technical skills, interests, social ties, and recency of project activity. To test the effectiveness of the we conducted an offline validation and solicited feedback from OSS contributors through interviews. While the results are encouraging (up to 63.0% hit rate for Top-20 projects), there is still room for improvement. We expect that additional signals, more sophisticated recommendation algorithms, and additional feedback cycles for contributors will lead to more effective recommendations in the future.

ACKNOWLEDGMENTS

We thank the OSS contributors who participated in the interviews for sharing great insights. We also thank Bobby Dresser, Evangeline Liu, Grace Vorreuter, Jenny Liang, and Mariam Guizani for their insight and feedback on this work. Nischal Shrestha conducted this work during an internship at Microsoft Research's Software Analysis and Intelligence in Engineering Systems (SAINTES) Group.

REFERENCES

[1] J. Wallen, "What happens when developers leave their open source projects?" retrieved April 5, 2022 from https://thenewstack.io/what-happens-when-developers-leave-their-open-source-projects/.

[2] C. Miller, D. G. Widder, C. Kästner, and B. Vasilescu, "Why do people give up flossing? a study of contributor disengagement in open source," in *Open Source Systems*, F. Bordeleau, A. Sillitti, P. Meirelles, and V. Lenarduzzi, Eds. Cham: Springer International Publishing, 2019, pp. 116–129.

[3] N. Eghbal, "Roads and bridges: The unseen labor behind our digital infrastructure," Tech. Rep., 2016, retrieved April 5, 2022 from https://www.fordfoundation.org/work/learning/research-reports/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure/.

[4] G. Assaf, M. Kumar, A. Kanyagia, and J. Jones, "Open source software in the social sector: Examining barriers, successes, and opportunities," April 2020, retrieved May 11, 2020 from https://socialimpact.github.com/#report.

[5] "Digital public goods," retrieved April 5, 2022 from https://digitalpublicgoods.net/explore/.

[6] J. T. Liang, T. Zimmermann, and D. Ford, "Towards mining oss skills from github activity," in *IEEE/ACM International Conference on Software Engineering (ICSE NIER)*, 2022.

[7] ——, "Understanding skills for oss communities," in *ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 2022.

[8] M. Guizani, T. Zimmermann, A. Sarma, and D. Ford, "Attracting and retaining oss contributors with a maintainer dashboard," in *IEEE/ACM International Conference on Software Engineering (ICSE SEIS)*, 2022.

[9] B. Trinkenreich, M. Guizani, I. S. Wiese, T. Conte, M. Gerosa, A. Sarma, and I. Steinmacher, "Pots of gold at the end of the rainbow: What is success for open source contributors," *IEEE Transactions on Software Engineering*, pp. 1–1, 2021.

[10] G. A. A. Prana, D. Ford, A. Rastogi, D. Lo, R. Purandare, and N. Nagappan, "Including everyone, everywhere: Understanding opportunities and challenges of geographic gender-inclusion in oss," *IEEE Transactions on Software Engineering*, 2021.

[11] M. Robillard, R. Walker, and T. Zimmermann, "Recommendation systems for software engineering," *IEEE Software*, vol. 27, no. 4, pp. 80–86, 2010.

[12] DeepSource, "Good first issue," retrieved April 5, 2022 from https://goodfirstissue.dev/.

[13] Mozilla, "Codetribute," retrieved April 5, 2022 from https://codetribute.mozilla.org/.

[14] S. Hanselman and K. Dodds, "First timers only," retrieved April 5, 2022 from https://www.firsttimersonly.com/.

[15] S. Balali, U. Annamalai, H. S. Padala, B. Trinkenreich, M. A. Gerosa, I. Steinmacher, and A. Sarma, "Recommending tasks to newcomers in oss projects: How do mentors handle it?" in *Proceedings of the 16th International Symposium on Open Collaboration*, ser. OpenSym 2020. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3412569.3412571

[16] H. S. Qiu, Y. L. Li, S. Padala, A. Sarma, and B. Vasilescu, "The signals that potential contributors look for when choosing open-source projects," *Proc. ACM Hum.-Comput. Interact.*, vol. 3, no. CSCW, nov 2019.

[17] Y. Hu, J. Zhang, X. Bai, S. Yu, and Z. Yang, "Influence analysis of github repositories," *SpringerPlus*, vol. 5, no. 1, p. 1268, 2016.

[18] A. S. Badashian and E. Stroulia, "Measuring user influence in github: The million follower fallacy," in *Proceedings of the 3rd International Workshop on CrowdSourcing in Software Engineering*, ser. CSI-SE '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 15–21.

[19] C. Liu, D. Yang, X. Zhang, B. Ray, and M. M. Rahman, "Recommending github projects for developer onboarding," *IEEE Access*, vol. 6, pp. 52 082–52 094, 2018.

[20] L. Zhang, Y. Zou, B. Xie, and Z. Zhu, "Recommending relevant projects via user behaviour: An exploratory study on github," in *Proceedings of the 1st International Workshop on Crowd-Based Software Development Methods and Technologies*, ser. CrowdSoft 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 25–30.

[21] Y. Huang, D. Ford, and T. Zimmermann, *Leaving My Fingerprints: Motivations and Challenges of Contributing to OSS for Social Good*. IEEE Press, 2021, p. 1020–1032.

[22] "Github rest api," 2021, retrieved September 20, 2021 from https://docs.github.com/en/rest.

[23] G. Gousios and D. Spinellis, "Ghtorrent: Github's data from a firehose," in *IEEE Working Conference on Mining Software Repositories*, 2012, pp. 12–21.

[24] "Github linguist," 2021, retrieved April 5, 2022 from https://github.com/github/linguist.

[25] M. Gerosa, I. Wiese, B. Trinkenreich, G. Link, G. Robles, C. Treude, I. Steinmacher, and A. Sarma, "The shifting sands of motivation: Revisiting what drives contributors in open source," in *IEEE/ACM International Conference on Software Engineering (ICSE)*, 2021, pp. 1046–1058.

[26] "Ovio explore projects," retrieved May 11, 2022 from https://ovio.org/projects.