Automatically mapping organizational structure for millions of open-source projects

Curtis Atkisson1 and Zixuan Feng2, Feng advisor, Atkisson mentor - Co-first authors
1: UMass Amherst
2: Oregon State University

Abstract: As Open-Source Software has proliferated, nonprofit Foundations dedicated to supporting established and nascent projects have arisen. These Foundations have different procedures for their projects and require that their projects follow different rules. These procedures and rules may impact the behavior of projects in many ways: projects may behave in a way to appeal to a certain Foundation, they may change which Foundations a project considers, and they may impact how work is done upon joining a Foundation. Current procedures for establishing which projects are a part of which Foundation are time-consuming and prone to errors. We develop a method that allows us to automatically categorize repositories as belonging to a list of important OSS Foundations using the World of Code data and an LLM for named entity recognition. From this, we are able to automatically categorize millions of repositories for which Foundation they belong to (if any). We compare our categorization to a ground truth sample of repositories that are known to belong to Foundations and get an F1 score of 0.91. These data have been made available as a regular part of World of Code as a P2F (Project to Foundation) mapping as well as through the MongoDB interface. We use these data to show differences on several key project characteristic variables for projects within and without Foundations, as well as across different Foundations. These maps will allow researchers interested in the entire OSS ecosystem to determine the effect of Foundation membership on important processes.

---

1. Introduction

Many OSS projects are based within OSS nonprofit foundations, which house many projects. GitHub allows for project repositories under organizational accounts. Many OSS Foundations have such accounts. For those Foundations with projects organized under their Github organizations, it is easy to label those projects as belonging to a

Foundation. However, a significant number of projects are not listed under their respective organizations on GitHub, creating a gap in understanding which project belongs to which entity.

When researchers investigate projects under different organizations, they often have to consult the organizations' materials directly and manually review and catalog the projects listed on GitHub. This approach from several deficits. Practically, this method is quite time- and labor-consuming, sometimes requiring months of a student's labor. Scientifically, such datasets may be biased, as many projects may be in the incubation stage or still in the process of being organized but not represented in the materials of the organization. This situation presents a particular challenge in investigating organizational transitions, as these projects might already be under governance or attempting to adhere to the proposed procedures of their respective organizations while researchers mistakenly think they are not. This lack of clarity creates significant challenges for researchers and foundations who aim to investigate areas such as organizational contribution patterns, governance structures, and onboarding processes within various organizational governance models.

Currently, there is no dataset that maps projects to their corresponding organizations. Therefore, the goal of this project is to create a dataset that categorizes each project according to its associated foundation. This will establish a foundational dataset for future analysis, such as examining specific governance models or contribution patterns within different organizations. We use methods that allow us to get the ground truth for project membership in organizations paired with methods that allow us to infer the organizational status of projects, which we validate against the ground truth.

2. Methodology

2.1 Sampling and classification performance:

Currently, there are around 100 million projects on GitHub. With a population size of 100 million projects, a confidence interval of 95%, and a 5% margin of error, the sample size is 385 projects. Thus, to evaluate the performance of the LLM model, we extracted 385 projects, half of which

are organizational projects, and the other half are non-organizational projects.

We first aimed to select a sample of repositories from GitHub that are not managed by organizations. To achieve this, we implemented a random sampling technique. This was done by iterating over each repository in our dataset (denoted as 'repos'). For each repo, we checked whether it was not associated with an organization. This check used a conditional statement: `if not repo['owner']['type'] == 'Organization'`. This statement filters out repositories where the 'type' of the 'owner' is not labeled as 'Organization'. In other words, it selects only those repositories that are owned by individual users or entities other than formal organizations.

We then picked the ground truth projects that are under organizations. We went through a list of 101 organizations and picked those with more than one project under their organization [2]. Next, we examined the repositories to identify sponsored projects, not just repositories for information. For instance, under the Linux Foundation GitHub organization (https://github.com/orgs/linuxfoundation/repositories), none of them are actually sponsored projects but are tools for the Foundation or information repositories. From this, we identified six foundations that contain sponsored repositories under their organization on GitHub. We know, then, that the repositories under those organizations are definitely part of the Foundation. In total, we have extracted 2,288 repositories from six different organizations. From these we select half of the 385 project sample.

2.2 Using Large Language Model (LLM) for Inferring

Based on data from FLOSS foundations, there are 101 organizations within the Open Source Software (OSS) communities [1]. Our focus specifically targets independent, international, and transparent foundations that are dedicated to supporting the development of a specific set of software products. These foundations play a crucial role in the OSS ecosystem, providing vital support and governance for their respective software projects (Table 1).

#	Foundation Name

1	Apache Software Foundation
2	Cloud Foundry Foundation
3	Django Software Foundation
4	Document Foundation
5	.NET Foundation
6	Eclipse Foundation
7	Fintech Open Source Foundation
8	FreeBSD Foundation
9	F# Foundation
10	Gentoo Foundation
11	GNOME Foundation
12	Kuali
13	Mozilla Foundation
14	NetBSD Foundation
15	NLnet Labs Foundation
16	Open Source Geospatial Foundation
17	OpenBSD Foundation
18	OpenSourceMatters
19	OpenStack Foundation
20	OpenStreetMap Foundation
21	Parrot Foundation
22	Plone Foundation
23	Python Software Foundation
24	Sahana Foundation

25	The Perl Foundation
26	Wikimedia Foundation
27	X.Org Foundation LLC

To determine whether a project belongs to one of these 27 organizations, we utilized a Large Language Model (LLM) for inference. Our method aimed to classify projects into these organizations in a simple yet effective manner, especially considering the extensive number of GitHub projects that exist today. Initially, we considered analyzing all ".md" files from each project, as most organizations require these files, particularly "readme.md" and "license.md". However, we found that analyzing all .md files was not necessary. The readme file, required by all organizations, often explicitly or implicitly mentions details about the organization, platform, software type, or communication methods. This information is typically indicative of the organization to which the project belongs. Consequently, we opted to use only the readme file for our analysis. We tested this approach by examining readme files from five different repositories.

The LLM successfully identified the correct organization for all five projects—the prompt we used:

"Based on the following README file, can you identify which organization or open-source software (OSS) community this project belongs to (from the list I give you)? It would be good to have a hieratical structure to show the community, organization, project, and maybe affiliation. Also, please list the communication channels mentioned in the file. I would like the information structured for a CSV file output, with each piece of information clearly defined and separated for easy future reference. List:...:"

3. Preliminary Findings

During the hackathon, in addition to familiarizing ourselves with the WOC (World of Code) [1] and designing the approach for analysis, the goal within these two days is to prove concepts. Based on the 385 repositories needed to validate the performance of the LLM (Language Model) models, each researcher analyzed half of the repositories they were randomly assigned to and manually went to each repository's readme file, copying and pasting

the readme file with the designed prompt questions to ChatGPT. In total, we have manually finished 107 repositories with the help of two researchers. The results are shown below:

	Predicted postive	**Predicted negative**
Actual postive	46	3
Actual negative	6	47

Permormance
Recall = 0.9388
Precision = 0.8846
F1 Score = 0.9105
Accuracy = 0.9118

4. Challenges

4.1 Scoping repositories from millions of projects.

Currently, GitHub has more than 500 million repositories. The first step before classifying the repositories as organizational or non-organizational is to narrow down the scope of repositories to which we should apply our analysis method. Currently, there is no detailed threshold specifying the level of activity required for organizational OSS projects. For instance, how many commits should they have, how many contributors should be involved, or how many authors should they contain? Additionally, it would be overly restrictive to simply apply a fixed threshold to filter out repositories, as some of the organizational projects are still in their early stages, with only a few contributors and limited commit activity. Furthermore, some of these projects serve as side projects for their main OSS repositories. Thus, we have applied a simple filter to exclude certain “private projects” that lack collaboration activities. The filters we have applied include: 1. No readme files (by default); 2. Number of commits per month < 1; 3. Contributors < 2.; 4; Contribution blob < 1.5; Forks < 1.

4.2 Differentiate between OSS tools that the organization uses and projects that the organization undertakes.

Another challenge we encountered pertains to the limitations of our classification results. We've come to understand that some of the organizational OSS projects are not traditional projects; rather, they are tools used by organizational OSS projects. Despite this, these tools remain open for collaboration and are not marked as "private." Consequently, these projects are included in our datasets. However, we believe that by applying the filters we've defined, we have already excluded some non-collaborative tools that organizations may have, if indeed there are any such tools that are not open for collaboration.

5. Future Work

As for future work, the first step is to complete the application of our method and finalize the analysis for the entire World of Code database. Once we have completed this phase, our plan is to conduct analysis between organizational repositories and non-organizational repositories. This analysis will include examining governance patterns in both types of repositories, highlighting the distinctions between them.

References

[1] Ma, Yuxing, et al. "World of code: an infrastructure for mining the universe of open source VCS data." 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR). IEEE, 2019.

[2] Izquierdo, Javier Luis Cánovas, and Jordi Cabot. "A Survey of Software Foundations in Open Source." arXiv preprint arXiv:2005.10063 (2020).