

Investigating Detection Methods for Cross-Agency Code Use

By: Evan “Taylor” Yates and Justin Gosses

Org: NASA OCIO Transformation & Data Division (TDD) Open Innovation Program

Date: 5/12/2020

Why Quantify Reuse of Government Code?

Congress requires Government agencies to publish a percentage of their code as open source, and for good reason. Open-source software promotes high-quality government code, maximizes code utility to the public, and saves taxpayer dollars by encouraging inter-agency code reuse. While few would question the benefits of open-source software, it is difficult to quantify those benefits in a meaningful way.

There are three main inhibitors when attempting to quantify the benefits of open-source software: (1) there is no mandatory mechanism to reliably track public software back to its original creator, (2) a significant amount of “users” that reuse government code are bot accounts that do not self-identify, and (3) there is ambiguity when deciding what qualifies as “reuse”. While we do not claim to have solved any of these problems directly, we are able to use alternative inference methods to minimize their impact on our results.

The goal of the project is to approximate how many government-sourced code repositories are being reused by other federal agencies. While the project was originally intended to be a NASA-specific tool, the work could benefit any government agency looking for feedback on its public codeshare practices. This article lays out the project’s overall structure and highlights some areas that can be improved, with the hopes that other interested agencies will provide feedback and ideas for future development.

Definitions

This article uses several terms that are specific to collaborative development and the “git” version control software:

- **GitHub** - *A popular public code-hosting platform that supports the “git” version control system.*

- **Application Programming Interface (API)** - A way for your code or application to exchange information with another application.
- **Repository (Repo)** - The location where you store a project's source code and related documents. A typical user will have multiple repositories.
- **Organization (Org)** - A collection of repositories and users that does not belong to any one individual, but instead to a group of administrators that the company or organization has appointed.
- **Commit** - A small change to a repository's code by a user who has permission to edit that repository.
- **Clone** - A direct download of a repository. No GitHub account is required. The identity of a user who clones a repo is not recorded.
- **Fork** - A copy of a repository that is placed in the user's personal repository collection on github.com. Users must have a GitHub account to fork repositories.

Overview

The scope of this project is limited to github.com. GitHub is commonly used by government agencies to release their code to the public. GitHub also provides a simple, free-to-use API that allows users to extract useful metadata on public GitHub assets such as repositories, organizations, forks, and even users. This wealth of metadata is an effective resource we will continually draw upon as we build out the project.

Defining Code Reuse (and what statistics should not be used)

While there are several metrics that can be considered when evaluating code reuse, there are two primary contenders: clones and forks.

In theory, clones are the best proof of software reuse since they are direct downloads of the original repository. However, there are two notable problems with clone data. The first is that there are many robot accounts that clone repositories automatically. These bot accounts are often difficult to identify and will ultimately dilute the result data. The second problem is that users can only see cloning data from the past two weeks, and even that data is extremely limited.

Forks, on the other hand, offer significantly cleaner and more useful metadata for the project. Most importantly, Forks are less commonly used by robots and can be tracked over any period. The downside of using forks is that a subset of the users who have reused a repository's code will not be considered. This is because some users will not bother to fork the repository but will instead just clone it. However, the "forking" subset will have lots of metadata that can be used to generate more robust results.

Defining an Agency User

The main problem to solve for this project is identifying GitHub users who are members of government GitHub organizations. Some organization members are publicly listed, but the majority are kept private. For those private users, we need to create some method of inferring the affiliation with their government organization.

The first step in identifying government-employed GitHub users is to locate known government GitHub organizations. Fortunately, a curated list has already been compiled and can be found [here](#). With this list, we can then retrieve metadata for public members in each organization. As mentioned above, this will likely miss a lot of users since most GitHub organizations leave members private by default.

To find the remaining private users, we can look at repository contributions. Repository contributions are often left public, even if the user who made the contribution is a private member of the organization. By combing through each public repository in a government organization, we can assemble a list of contributing users on those repositories along with the number of contributions each user made.

With this contribution data, we can then create a method of approximating the likelihood of a GitHub user's membership status in a government organization. To do this, we use a simple algorithm that incorporates the number of contributions a user has made on repositories belonging to an agency. As the number of contributions increases, the more likely the user is a member of that government organization. As a general heuristic, users with more than 100 contributions to an organization are considered likely members of that organization. The algorithm is intentionally conservative to preserve the quality of our result data.

Defining an Agency User Who Reuses Code from Another Agency

Once we have a list of weighted government GitHub members ("weighted" meaning each user has a score indicating how likely they are to be a member of a government agency), we can cross reference the list with known NASA repository forkers.

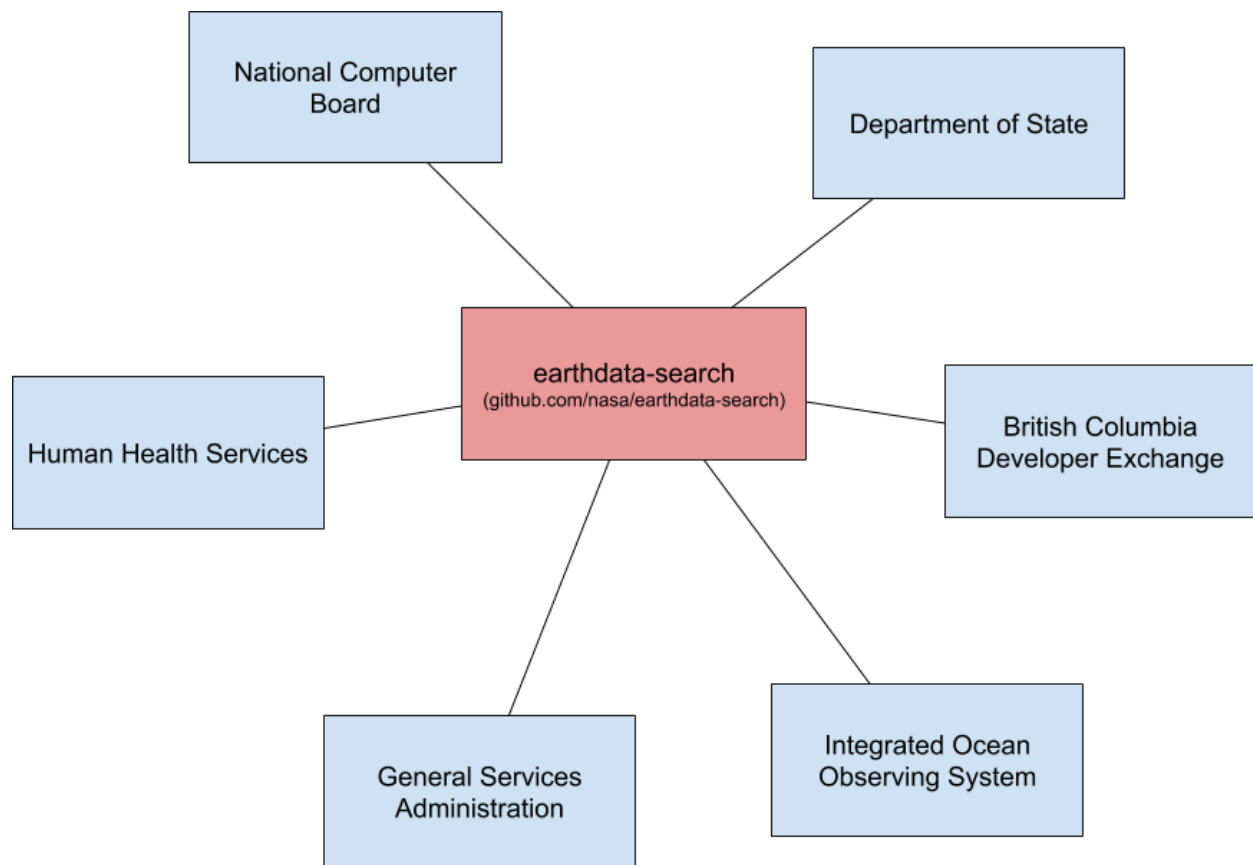
Compiling the list of forkers is relatively simple. The GitHub API provides an option for retrieving a list of all the existing forks of a repo. From this list of forks, we can pull each fork owner ("forker") and store them in a separate list. We repeat this process for each NASA public repository until we have a full list of NASA repo forkers. Before we can move to the next step, we need to remove any users who are known NASA members. The goal of the project is to find users who are members of other government GitHub organizations, so including our own members will only reduce the quality of the data.

Now that we have our lists (NASA forkers, and weighted government org members), we can cross reference the two, looking for usernames that show up in both lists. At this point, we have completed the minimum requirements of what we originally set out to do. We have a resulting list of users who have a) forked one or more NASA repos and b) made contributions to repos belonging to other known government GitHub organizations.

Just because we meet the minimum requirements for our project does not mean we have meaningful result data. Each user in the result list is weighted, as mentioned before, meaning they have a score ranging from 0.0-1.0. Our list includes users with any score, if they have made 1 or more contributions on government repos. To refine our data, we need to set a threshold for the minimum allowed score. This value does not have to be set in stone, but we recommend a threshold of at least 0.5.

Sample Results

The diagram below shows a sample of the type of insights we can derive from our result data. In the diagram, the central red node is one of the most commonly shared public NASA repositories amongst other government agencies. The adjacent nodes represent the various agencies or government organizations that have been identified as likely benefactors of the NASA repository. For this representation, we used a threshold value of 0.5.



What You Can Provide?

Since this project is still very young, there are several areas that can be improved along with features that could be added.

Private Member Prediction

Currently, our only metric used when inferring private members of a government organization is the number of contributions made. We'd like to develop a small machine-learning model that takes into consideration a wider range of features when making this prediction. For the model to be effective, we will need more training data than the NASA GitHub organization alone can provide. If you have the authority to disclose private members of your GitHub organization (with members' consent), we'd greatly appreciate the additional data to train our model. We will include contact information at the end of the article for any inquiries or feedback.

Refine “Reuse” Definition

Our project currently associates forks with “reuse” and does not go much further. Ideally, each fork would be analyzed to see how much of the repository was actually forked. If the fork happened when the original repo was new, then we would not consider that a “heavy” reuse case since only a small amount of code was actually carried over. Another factor to consider is how much a fork has diverged from the original repo. If a user forks the repo but never actually adds their own code to it, we would be less inclined to consider that fork an instance of reuse. These are just a couple of examples of ways to refine the concept of “reuse” in this project.

Package Project into an API

While there could be a good amount of interest in this project from other agencies, not everyone will want to download a code repository and run the analysis manually. A potential solution for this would be to build out the tool into an API. That way, an agency could easily incorporate their own results into a website or pre-existing application.

Contact Information

We expect there to be plenty of other use cases and potential features that could enhance this project and would greatly appreciate your feedback. Our contact emails are listed below:

- Evan “Taylor” Yates (evan.t.yates@nasa.gov)
- Justin Gosses (justin.c.gosses@nasa.gov)

Since this project is not yet approved for open source, we cannot share the code itself. However, we hope to be able to publish the project soon and will update the article when the code is released.