GitHub Repository Finder – A Sample Dataset

*Adapted from: Gebru, Morgenstern, Vecchione, Vaughan, Wallach, Daumeé, and Crawford. (2018). Datasheets for Datasets.**

1. Motivation

1.1 *For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.*

This dataset was created to illustrate the operation of GitHub Repository Finder (hereafter referred to as GRF), a function to search for GitHub repositories based on a determined keyword term and a timeframe.

However, basic GitHub users (without or with Basic Authentication Token) who want to crawl data from GitHub, especially search data, have reported several obstacles during the data extracting process. First, by default, the calls only return the first page of the search results, containing at most 100 results. Second, the Search API returns only up to 1000 results per query (including pagination). Third, limit rate restricts the number of API calls per minute (10 call/minute for unauthorized and 30 calls/minute for authorized basic users), thus, extracting process can be interrupted without notice.

GRF helps users overcome these obstacles by:

- Dividing search results into narrower time segments (by setting a parameter which determines how many hours per segment).
- Using dynamic pagination, breaking down each 4-hour search segment into smaller queries by manually calculating number of pages and fetching each page.
- Setting sufficient break time between each call to comply with GitHub rate limit rules.

With this dataset, we illustrate the result obtained from GRF. To obtain this dataset, we use keyword "python" and set a duration of 3 days to extract information of all repositories containing the specified term, created within 3 days before the extraction date

1.2 *Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?*

This dataset, together with the source code it represents, were created by Tolga Depecik, Trang Bui, Vasileios Syrpas, Janick Boekhorst, Iliana Despoina Chlimintza. The dataset and source code were created for an assignment of the courses Online Data Collections (oDCM) at Tilburg University. The source code is created under the instruction and coaching of Hannes Datta.

1.3 *Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.*

The creation of this dataset was not funded nor were there any associated grant.

* https://arxiv.org/abs/1803.09010

2. Composition

2.1 *What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of in- stances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.*

The data was collected from Github.com using API combined with web scraping. Each instance consists of a repository created by a GitHub user. These repositories are obtained from search query based on keyword "python" and a timeframe between date of extraction and 3 days before. There is only one type of instance.

2.2 *How many instances are there in total (of each type, if appropriate)?*

The number of instances (repositories) depends on the keyword and the number of days the user is interested in. In this case, the dataset contains a total of 4613 repositories based on the keyword "python" created between October 14, 2021 and October 17, 2021.

2.3 *Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).*

The dataset contains all public repositories available on GitHub based of the keyword and the determined extraction duration.

2.4 *What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a de- scription.*

A repository contains files of a project, and it gives the opportunity for multiple users to interact at the same time in the project. The available information inside of the repository includes Name of Repository, Stars, Forks, Creator, Contributors, Date of creation, Latest update, Language used, topic, link, Readme file...

For each repository, the following information was collected:

- id: identification number of the repository on GitHub
- name: name of the repositor
- URL: link to the repositor
- language: the programming language in which the source code is written.
- created: repository's date of creation
- stars: number of users who have starred (i.e., bookmarked or showed appreciation to) the repository
- watch: number of users who have watched (i.e., followed the changes of) the repository
- forks: number of users who have forked (i.e., created a branch for the repository in their own GitHub page) the repository
- readme: content of the readme file in the repository

Except for readme which was decoded to remove the image to be saved in a .csv file, all other data is raw.

2.5 Is there a label or target associated with each instance? If so, please provide a description.

Each GitHub repository has a unique identifier. In this case, the identifier is saved in id. Each id is a unique combination of eight numbers.

2.6 *Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.*

Yes. The missing information in the dataset is represented with "NaN". In some repositories, the missing information can be present in language and/or readme columns. This is because the repository is empty, or the creator did not create - upload any description for the project.

2.7 *Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.*

There are two relationships between instances: content and time. First, all individual instances are crawled based on the keyword that the user is interested in. In this case, the word "python" is related to all instances. Second, the extracted repositories were all created within the timeframe pre-specified by the extractor.

2.8 Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.

No.

2.9 *Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they ex- isted at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.*

The dataset is relative to the external website Github.com. The dataset will be available in the future, however, if a repository is removed or hidden by its creator, this individual instance will not be available anymore for future extraction. Other problem can be that GitHub might stop providing API access or changing the structure of the API response files. Users need to check on GitHub's API documentation website (https://docs.github.com/en/rest) for the latest changes and developments. To the best of our knowledge, there has been no there official archival versions of GitHub repository data.

2.10 *Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctorpatient confidentiality, data that includes the content of individuals non-public communications)? If so, please provide a description.*

Only public repositories were extracted. Thus, the information is available for everyone, as a result, the dataset does not contain confidential information.

2.11 *Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.*

The dataset only contains public information. The readme is user-generated content which is not controlled by GitHub. Thus, there can be the case that inappropriate content is visible in the dataset out of control of the extractors.

2.12 *Does the dataset relate to people? If not, you may skip the remaining questions in this section.*

The dataset is not relative to people.

3. Collection Process

3.1 *How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by sub- jects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.*

Each instance in the dataset is a repository created on GitHub during the focal timeframe. The data was directly observable on GitHub API link and repository detailed page. By specifying the search keyword and the timeframe in the search query, users can construct their search API URL. The search API URL gives information about the repository id, name, URL, programming language and repository statistics (number of stars, watchers, forks). The content of the readme file can be found in detailed page of each repository.

3.2 *What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software pro- gram, software API)? How were these mechanisms or procedures validated?*

Both API calls and web scraping were used to collect this dataset. We chose to use API first as it is the official way to extract data from GitHub and the results are provided in well-structured json files. However, as Search API does not provide readme data. We needed to scrape data using python and the BeautifulSoup package.

By calling search API from GitHub using python, we were able to collect the following variables: repository id (id), repository name (name), repository url (url), programing language (language), date of creation (created), number of stars/watch/forks (stars/watch/forks) of each repository.

As API does not provide content of the readme file for each repository, a web scraping function was written in python using BeautifulSoup package to access repositories' detailed page and retrieve the content of readme file, saved as variable readme.

Detailed explanation of the data extraction function (GRF) is documented in the function documentation available online at: https://github.com/thtbui/github-repository-finder

3.3 *If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?*

The complete dataset was retrieved without sampling. We were able to extract all repositories in the search query, i.e., information of all repositories containing the specified keyword term created within the timeframe.

3.4 *Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?*

The above-mentioned students are involved in this process. The collection process is part of the Online Data Collection and Management course at Tilburg University. No financial compensation is concerned. The students will be rewarded with 3 ECTs after finishing the course, providing the completion of this assignment and the final exam.

3.5 *Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the time- frame in which the data associated with the instances was created.*

Data was collected on October 16, 2021 at 12:42:59. It includes repositories created on Github from 2021-10-13 12:42:59 to 2021-10-16 12:42:59. The timeframe of each API call is documented in a log file saved as log.txt in user's working directory. This file provides information about the timeframe of each API call and its status. Users can use this file to monitor the data extraction. Furthermore, during the collection process, GRF also prints out the number of repositories extracted for each API call. If there is anything wrong during the extraction, an error message will also be print out, showing which call is failed.

3.6 *Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.*

No. This is not applicable as we use API provided by Github and only collecting public-facing content. Pulling public-facing content via API and compiling this type of content are allowed as stated in Github privacy statement (https://docs.github.com/en/github/site-policy/github-privacy-statement)

3.7 *Does the dataset relate to people? If not, you may skip the remaining questions in this section.*

No, the dataset only contains information about repositories.

4. Preprocessing, cleaning, labeling

4.1 *Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remain- der of the questions in this section.*

The only modification of the raw data is to remove the images from readme files so that the content of readme can be written into .csv file. Furthermore, to make readme content easier to read, string "n/" is replaced with space. Columns are labelled by setting a concise, meaningful, and clear name for each column. This results in a dataset with 9 columns: "id", "name", "url", "language", "created", "stars", "watch", "forks", "readme".

4.2 *Was the "raw" data saved in addition to the prepro- cessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.*

Yes. Each API call provides a .json file. All .json files are combined into one single file and saved in the user's working directory. The .json file for the sample dataset can be accessed at: https://github.com/thtbui/github-repository-finder

4.3 *Is the software used to preprocess/clean/label the instances available? If so, please provide a link or other access point.*

Yes. The source code used to label the variables is part of GRF, written in Python programming language, available at: https://github.com/thtbui/github-repository-finder . Instructions to execute Python scripts can be found at: https://www.python.org/.

5. Uses

5.1 *Has the dataset been used for any tasks already? If so, please provide a description.*

No.

5.2 *Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.*

Currently, no. However, if any analysis is done on this dataset, all documents related to the dataset can be found at: https://github.com/thtbui/github-repository-finder

5.3 *What (other) tasks could the dataset be used for?*

This specific dataset is used to illustrate the operation of GFR. Users can also use datasets generated from this function for different research purpose, such as text-mining, comparing repository statistics under different keywords, or comparing repositories over time.

5.4 *Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?*

Content of each repository is user-generated. Therefore, users need to take into account the dynamic changes of information such as name, stars, forks, watch, readme. Moreover, as mentioned above, users can delete or hide repositories from public, in that case, these data will be lost in future extraction.

Regarding the use of the GRF to extract data, in order to avoid uncompleted data extraction, users should consider the total number of repositories under each keyword to identify how large the segments should be divided. More details about this issue can be found in the function documentation.

5.5 *Are there tasks for which the dataset should not be used? If so, please provide a description.*

No.