# Data Collection Documentation (Draft)

Steven Deutekom

June 2019

# Contents

# 1 Introduction

The goal of this project is to collect source code samples from individual authors from various online sources. It requires finding samples that have certain sociolinguistic characteristics such as gender, region, and experience. In addition to sociolinguistic characteristics, it is necessary that the samples are the work of only a single author. Data that meets these needs is being collected and used to create a dataset with information on authors and their source code.

The collected data will be used for sociolinguistic research into how people use programming languages. Current and future University of Lethbridge students will use this research to learn how sociolinguistic characteristics affect how programmers write code. Previous research was conducted with a small dataset of student programs. This new dataset will contain many samples from a larger set of programmers.

The document is broken into three main sections. Each section details one of the sources that was used to collect data. First, the collection methods used to gather source code from GitHub. Then, the methods used to collect source code from Codeforces. Lastly, the methods that were used to add gender data to the samples collected.

Each section introduces the sources and collection methods. Then it discusses the pros and cons of the source. Next an overview of the process of collecting data is given, with diagrams to help visualize it. Following this a more in depth technical examination of the collection process takes place. Finally, some reflection on the source is given (needed?).

# 2 Collecting Source Code From GitHub

## 2.1 Data Sources

## 2.2 Collection Startegies

## 2.3 Pros And Cons

## 2.4 Process Overview

### 2.4.1 GhTorrent and Google Big Query

### 2.4.2 Collecting Git Projects

### 2.4.3 Collecting Git Commits

## 2.5 Process Details

### 2.5.1 GhTorrent and Google Big Query

**Process**

**Data**

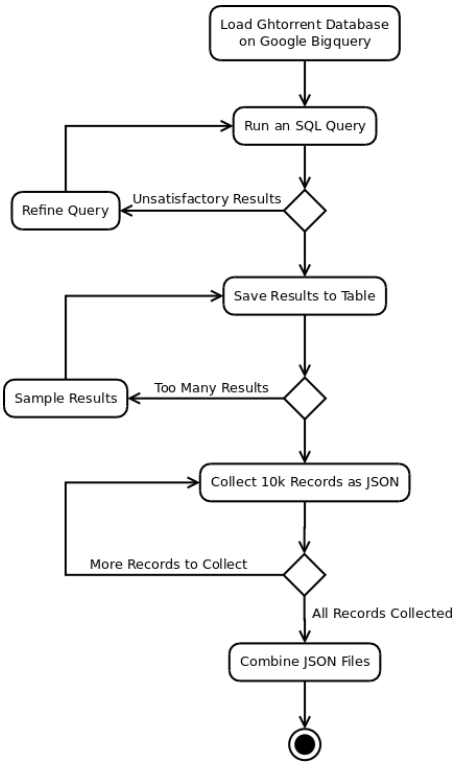


Figure 1: Getting Data From Ghtorrent

**Additional Issues**

### 2.5.2 Collecting Git Projects

**Process**

**Data**

**Additional Issues**

### 2.5.3 Collecting Git Commits

**Process**

**Data**

**Additional Issues**

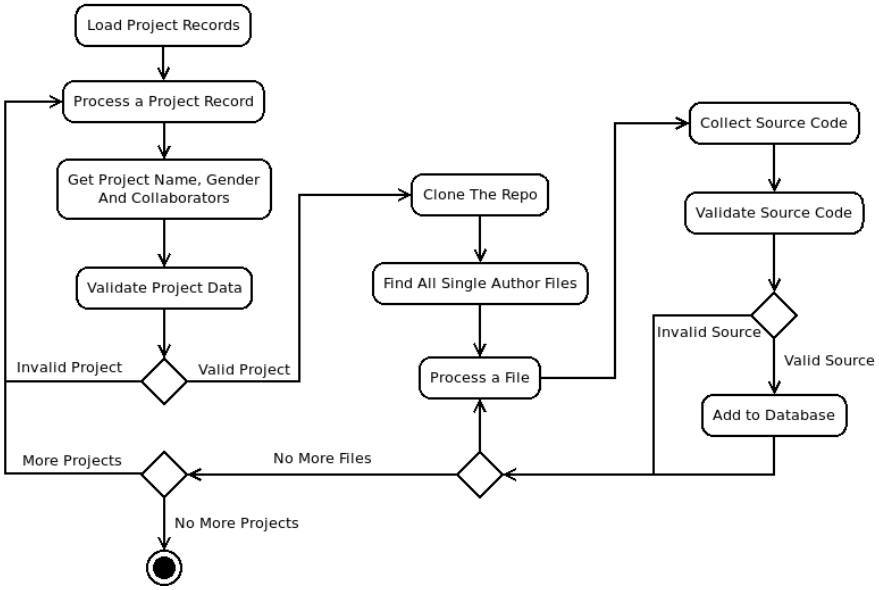


Figure 2: Getting GitHub Project Source Code

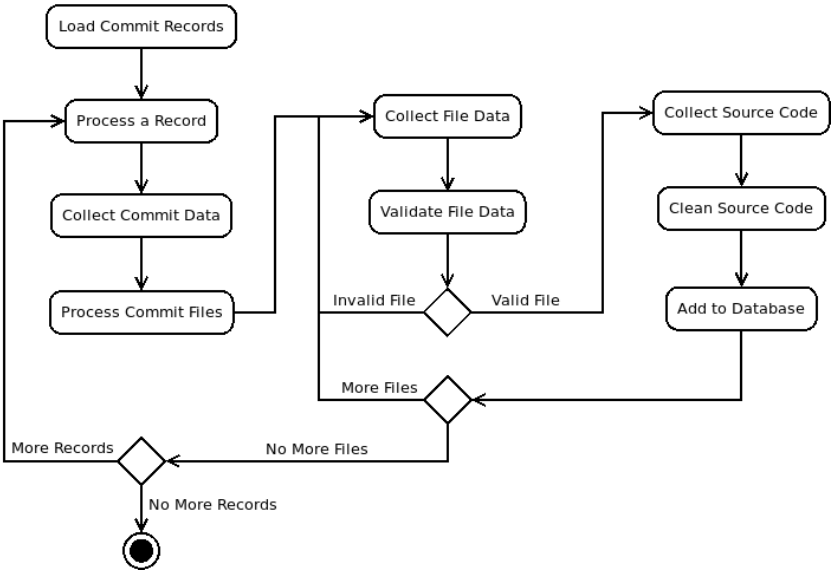


Figure 3: Getting GitHub Commit Source Code

# 3 Collecting Source Code From Codeforces

## 3.1 Sources

## 3.2 Pros And Cons

## 3.3 Process Overview

## 3.4 Process Details

### 3.4.1 Using Selenium

**Process**

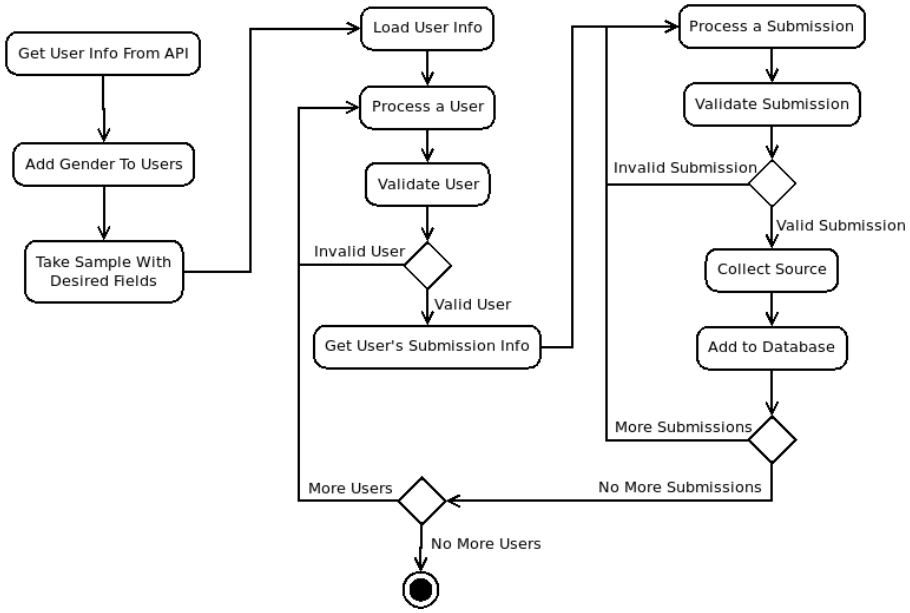


Figure 4: Getting Data From Codeforces

**Data**

**Additional Issues**

### 3.4.2 Using Links Only

**Process**

**Data**

**Additional Issues**

# 4 Adding Gender Labels

One of the sociolinguistic features that is important to the project is Gender. Unfortunately, gender is available for authors on Github or Codeforces. Since it cannot be collected along with author information gender must be obtained in other ways. This can be done using an authors first name. There are a number of websites and APIs available that offer gender data for first names. All the gender information for authors in the dataset were gathered in this way.

## 4.1 Sources

There are several websites and APIs for adding gender for first names. They include gender-api.com, genderize.io, namsor.com, and nameapi.org. This project primarily used genderize.io because it is free and allows 1000 name lookups per day. In addition to this a table was kept in the database to store the names that were already known. The service offers a 'male', 'female', or 'unknown' verdict when given a name. It also gives a probability that a result is correct (from sample?). In addition it is possible to pass a country code to along with the name to obtain results that are more specific? The other APIs all offer similar services, but they all require monthly payment for more than a small number of name guesses.

## 4.2 Pros And Cons

Inferring gender from names is not an easy task (some examples and sources). It is also not always very clear how a service is determining the gender (cite). So this is immediately a problem. The probability given can help to determine the likelihood of results being correct, but it still is only as good as the service. There are some stats given by (info and paper citation) that comment on the accuracy of the above 4 services.

Aside from the accuracy of these methods being suspect, the major con is that for the most part they are paid services. Some of them do offer free limits, but only genderize is really worth it for any larger needs, and even then only with an offline backup of already labeled names. The good news is that there are many people sharing a small number of names, so a large number of samples can be labeled with a small database.

People are not always truthful with their names on these profiles, sometimes they do not even give a name. As a result there are fewer samples to work with if gender is desired. If country is going to be used to add accuracy to the label then it lowers the number of samples available further.

It is possible though with several sources available that when a selection of samples is made they can have their gender labels confirmed by other sources. Because many people share a similar name it is also not as big a deal to use free credits or even pay for a small period to obtain better results for a set of good samples.

All services are less effective at labeling Asian names (citation). This can present a problem for datasets that are using large numbers of samples from Asian countries.

This is really the only way other than contacting authors directly to find out their gender. Which would be problematic with any reasonably sized database. Not to mention many people would be unlikely to respond to such a question.

It is also worth noting that these methods also are firmly gender binary. They do not take into account any other factors. And the connection between gender of a name and biological sex is by no means guaranteed.

It is not always clear wether a country should be used, though it is suggested that this improves the results (cite).

The number of samples can be low for a name so even if the probability is high it is not really certain if the sample size is large enough.

## 4.3 Process Details

### 4.3.1 Process

The process used for collecting gender for names is relatively simple. Given a name the database is checked to see if it has the name already. If the name is not in the database it is looked up with the genderize.io API. If the API returns a result that result is added with the name to the database. If the API limit has been reached then a appropriate(what?) response is given. The scripts all have options (Do they?) to save results for later processing if gender cannot be determined. If there name is able to be labeled then the label and the probability are returned.

The labels added to the scripts do not use country codes. As stated above it will be desireable to re-label the names on samples that are used by submitting the names and countries again to other APIs to see if results can be confirmed or refined. Because this will be done after it is likely that some authors will have been missed because they had unknown genders and so they were not added to the dataset. This may throw off the statistic of the samples a bit. The most important thing is to have enough good samples with reliable gender labels to use in experiments, so loosing a few users is acceptable. Better to have less data that is high quality than to have more data that is lower quality.

### 4.3.2 Data

The data consists of just the gender and the probability the gender label is correct. The gender is a string 'male', 'female', or 'unknown'. The probability is a float between 0.0 and 1.0. The Case of a name is ignored (is it?). There is also a 'count' field that is returned that is described as 'the number of entries examined to calcuate the result' (cite). and if country code is given the country code is also part of the response. Also the name is part of the response.

# 5 Conclusions