Investigating the Validity of Ground Truth in Code Reviewer Recommendation Studies

Emre Dogan

Bilkent University

June 17, 2019

Outline

1 Introduction

2 Code Review Process

3 Ground Truth

4 Example Scenario

5 Problem Definition

6 Review Selection
- Review Selection in Real Life
- Review Selection in Recommendation Models

7 Quantitative Evidence for the Ground Truth Problem

8 Solution Alternatives

9 References

Introduction

- **Code Review:** An inspection of a code change by an independent third-party developer in order to identify and fix defects before an integration to improve software quality[**?**].
- It is a challenging task to find the ideal reviewer for a changeset.
- There are several studies to recommend the ideal reviewer for a changeset:
  - Heuristic Based Approaches
  - Machine Learning Approaches
  - Graph Based Approaches
  - Hybrid Approaches

Code Review

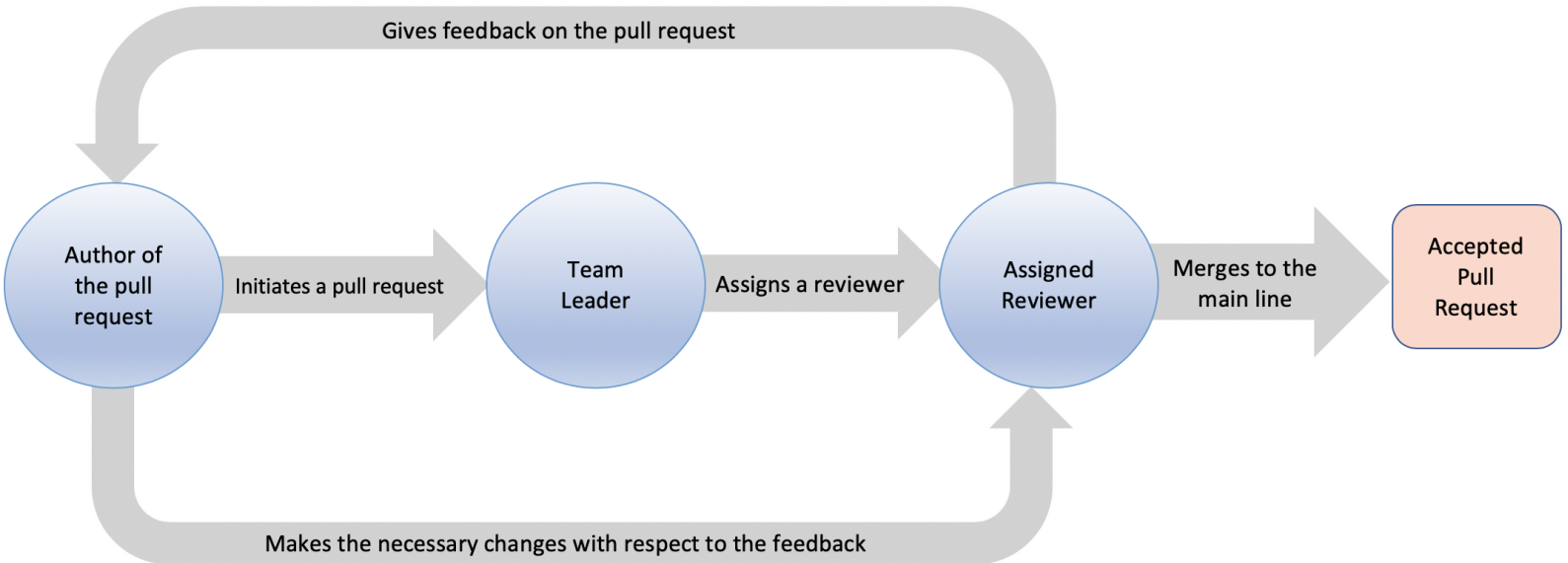


Figure 1: A typical code review scenario in real life.

Ground Truth

Definition

Ground Truth: Ideal output label of an algorithm.

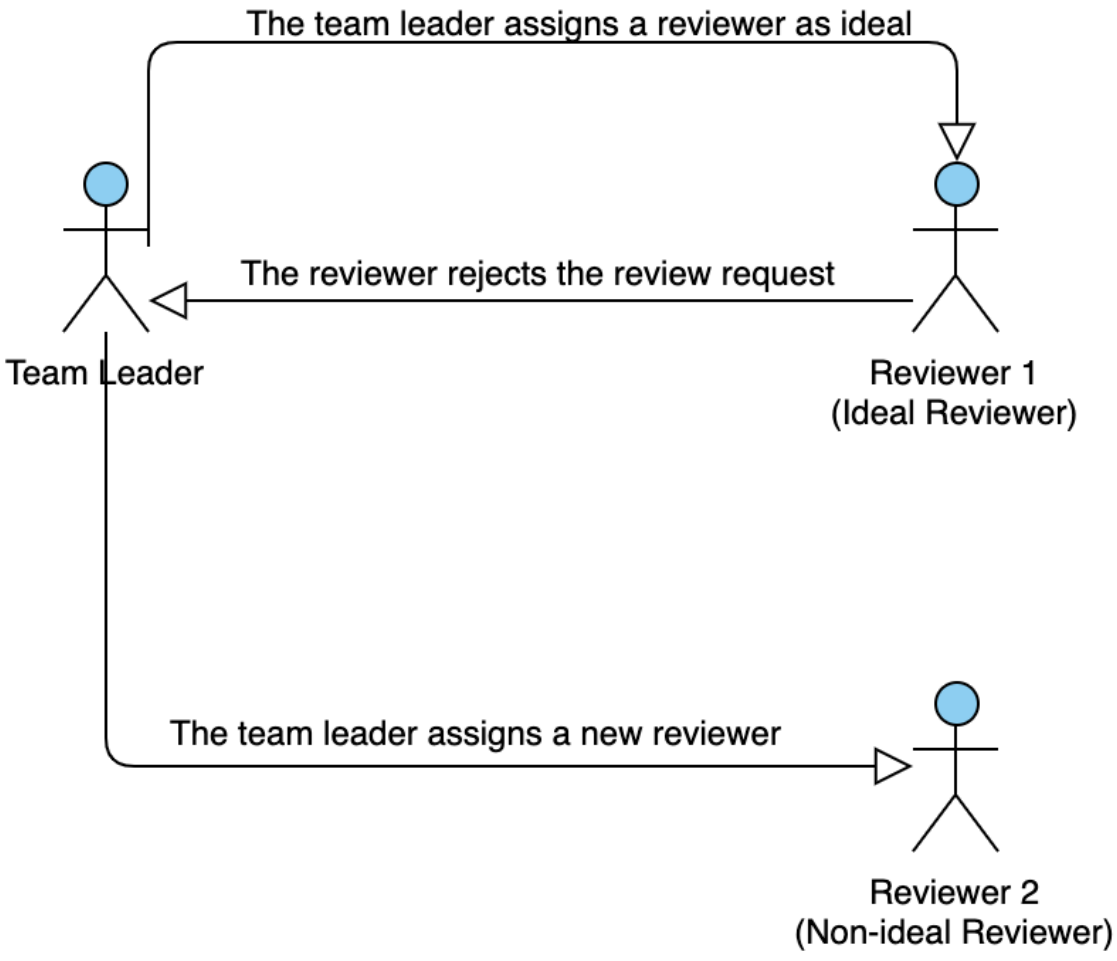
Ground Truth in Software Engineering

The more human aspects involved, the more tendency to the ground truth problems.

Ground Truth in Code Reviewer Recommendation Process

- Recommendation studies rely on the real-life reviewer assignments.
- These studies assume that these assignments are ideal.
- Studies in real-life projects (OSS or proprietary) show that code reviewers are not assigned with the aim of finding the ideal one.

Example Scenario



Problem Definition

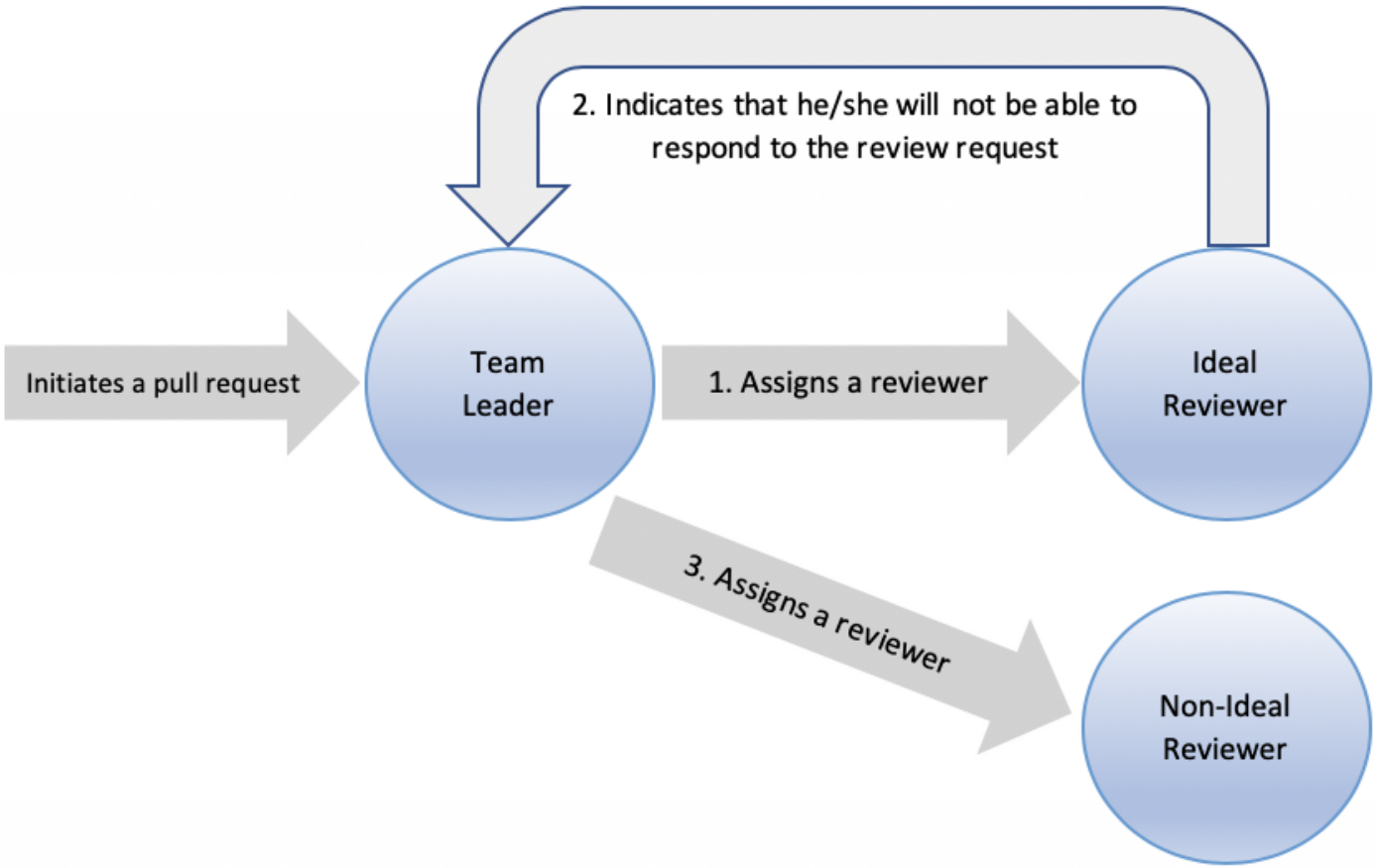
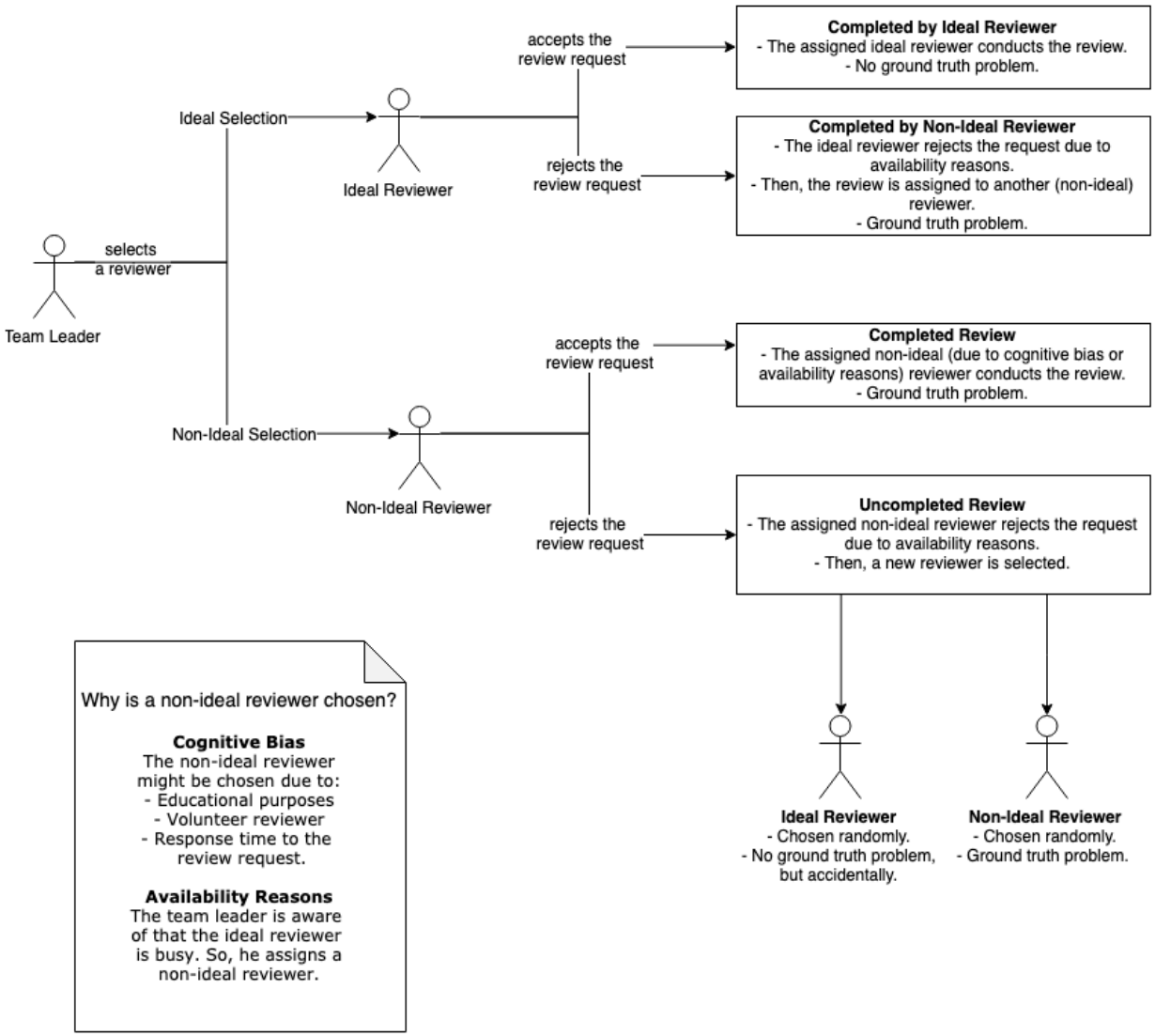


Figure 3: Ideal Reviewer Selection Problem.

Problem Definition-II



A Set of Real-Life Scenarios

Pull Request	Actual Reviewer	Ideal Reviewer	Recommended Reviewer	Case	Correctness of Algorithm w.r.t. Actual Reviewer	Correctness of Algorithm w.r.t. Ideal Reviewer	Ground Truth Validity
1	John	John	John	r=a and a=i	✓	✓	Yes
2	John	Mary	John	r=a and a≠i	✓	X	No
3	John	Mary	Mary	r≠a and a≠i	X	✓	No
4	John	John	Mary	r≠a and a=i	X	X	Yes
5	John	Mary	James	r≠a and a≠i	X	X	No

Table 1: Real-life scenarios to illustrate the ground truth problem.

r: recommended reviewer, a: actual reviewer, i: ideal reviewer

Notice

Pull Requests 2, 3 and 5 have a problematic ground truth definition.

How Reviewers are Selected in Real Life

Related Studies	Considered Method
[?, ?, ?]	Reviewer experience
[?, ?]	Code familiarity
[?, ?]	Patch characteristics
[?, ?]	**Volunteer for review**
[?, ?]	**Workload**
[?]	**Physical Proximity**
[?]	**Availability**
[?]	**Response time to review requests**
[?]	**Training the new hires**

Table 2: Real-life factors affecting reviewer selection.

Recommendation Models

APPLIED METHOD
Bayesian Network
Change History of Source Code Lines
Decision Tree
Expertise in Related Technologies
Genetic Algorithm
Information Retrieval
K-nearest neighbors (KNN)
Latent Dirichlet Allocation
Latent Factor & Neighborhood
Path Similarity
Previous Review Success
Random Forest
Social Network Analysis
Support Vector Machines
Text Similarity of PR Requests

Table 3: Automated Reviewer Recommendation Methods

Comparison of Real Life and Recommendation Studies

Hypothesis

Reviewer selection process takes place differently in real life and recommendation models.

Quantitative Evidence

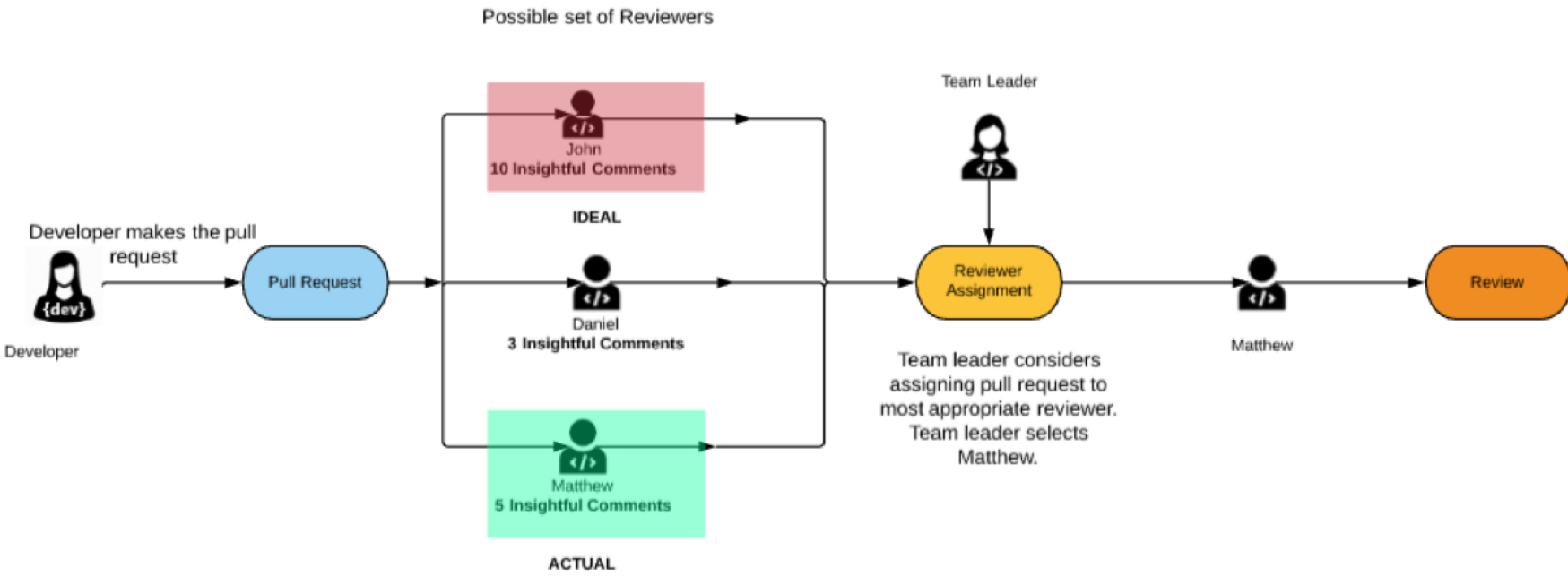
Project Name	Total Number of Pull Requests	Number of PRs with at least one non-responsive reviewer	The ratio of PRs having at least one non-responsive reviewer
Android	36,771	24,367	66%
LibreOffice	18,716	3,039	16%
Open Stack	108,788	24,589	23%
Qt	65,815	30,630	47%
TOTAL	230,090	82,625	36%

Table 4: PR Analysis of 4 Large OSS Projects [**?**].

Solution Alternatives

- Expensive Setup in Real Life
- Forward-Looking Mining

Expensive Setup in Real Life

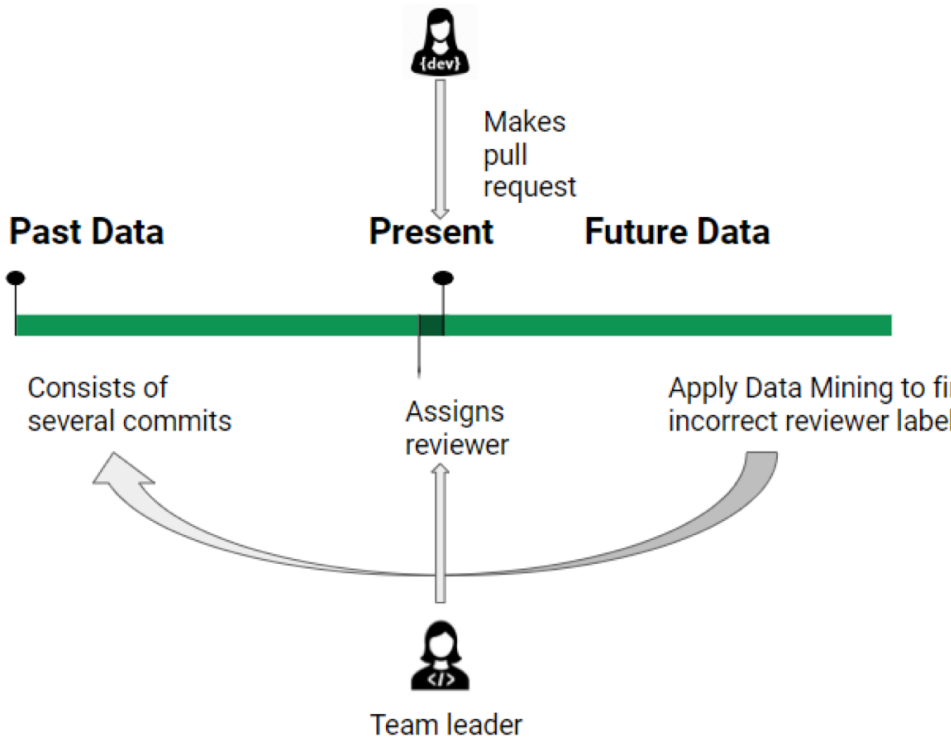


Expensive Setup in Real Life

i. To fix an issue, developer implements a solution and commits this solution as a pull request.
ii. Team leader assigns every possible reviewer in the team as a single reviewer to the pull request.
iii. Each reviewer reviews the request and provides comments.
iv. Team leader or the developer selects the ideal review and reviewer based on metrics (i.e. providing insightful comments in the shortest time possible) among the reviewers and reviews.

Forward-Looking Mining

- If a bug is reopened, it is a potential indicator that the assigned reviewer was not the ideal reviewer for that pull request.
- Deleting these instances will increase the validity of the dataset.



References