Reproducibility Report for ACM SIGMOD 2020 Paper: “BugDoc: Algorithms to Debug Computational Processes”

SHIMIN CHEN, Institute of Computing Technology, Chinese Academy of Sciences

The reproducibility run generated data and results corresponding to Figures 2–4 in the SIGMOD paper, which show similar behavior to the figures in the paper. This confirms the experimental results in Sec 5.1 Synthetic Pipelines in the paper.

1 INTRODUCTION

This reproducibility report is for the following paper:

- Raoni Lourenço, Juliana Freire, Dennis Shasha (New York University). *BugDoc: Algorithms to Debug Computational Processes.* SIGMOD 2020.

Section 5 experimental evaluation in the paper consists of three subsections: 5.1 Synthetic Pipelines, 5.2 Scalability, 5.3 Real-World Pipelines.

- The reproducibility run recreated Figures 2–4 in Section 5.1 Synthetic Pipelines, showing results similar to the figures in the paper.
- The reproducibility run did not generate figures 5–6 in Section 5.2 Scalability, or Figure 7 in Section 5.3 Real-World Pipelines.

The recreated Figures 2–4 confirm the main finding in the paper that the proposed methods in the paper outperform state-of-the-art methods in precision, recall, and F-measure.

2 SUBMISSION

The reproducibility submission contains information on source code, datasets, hardware, and experimentation in a one-page description (as shown in Appendix A). We describe each item in the following:

- Github repository with code and scripts: `github.com/VIDA-NYU/BugDoc/blob/sigmod20/`. The clone command: `git clone --branch sigmod20 git@github.com:VIDA-NYU/BugDoc.git`.
- Detailed readme file: `https://github.com/VIDA-NYU/BugDoc/tree/sigmod20#2-experiments`
- Datasets: Datasets are synthetically generated in the experiments.
- Hardware: Since the experiments mainly use Python, Java, and PostgreSQL, any Unix-like environment with necessary libraries can support the experiments.

3 HARDWARE AND SOFTWARE ENVIRONMENT

Table 1 compares the hardware and software environment described in the reproducibility submission and used in the reproducibility run.

Please note that the experimental machine in the paper is different from the description in the reproducibility submission. Section 5 in the paper says “All experiments were run on a Linux Desktop (Ubuntu 14.04, 32GB RAM, 3.5GHz × 8 processor)”. As shown in Appendix A, the submission indicates “the experiments were executed on a MacBook Pro, running macOS Catalina version 10.15.7”. As the latter provides detailed hardware information, we list the latter in Table 1.

As shown in Table 1, the machine used in the reproducibility review has similar hardware configurations as the machine described in the reproducibility submission. Moreover, we set up the software environment to be as closely as possible to the description in the submission.

Table 1. Hardware and software environment

	Reproducibility Submission	Reproducibility Review
Machine/OS	MacBook Pro, macOS Catalina 10.15.7	Linux Desktop, Ubuntu 18.04
CPU	2.2GHz Quad-Core Intel Core i7	3.4GHz Quad-Core Intel Core i7-4770
RAM	16GB	32GB
Storage	HDD	HDD
PostgreSQL	version 11	version 11
Python	version 3.6	version 3.6.9
Java	version 1.8.0_144	version 1.8.0_282

4 REPRODUCIBILITY EVALUATION

4.1 Process

We followed the reproducibility submission to set up and run the experiments. The main process is automated by a number of scripts. We addressed the following challenges with the help of the authors:

- We found that the script had been running for a week, which could mean something was wrong. Therefore, we contacted the authors to understand whether this was normal or not. It turned out that the experiments could run for 10–20 days. *A lesson learned is that the reproducibility submission should give an estimate of the run time if it is longer than a threshold (e.g., 24 hours).*
- The first run of the experiments produced the sub-figures in Figure 2 only. The other figures were missing. We contacted the authors to understand if this was expected or not. With a few communications, the authors fixed the problem and updated the github repository. *A lesson learned is that the reproducibility submission should specify which figures in the paper are expected to be generated.*
- We saw that two of the seven bars were missing from the produced figures. We fixed the problem with the help of the authors. It turned out that a PostgreSQL language package should be installed. This information was missing from the submission description, but it was in a readme file in the repository. Since we mainly followed the submission description, we overlooked the latter in the initial run. *A lesson learned is that it would be nice if the submission can list all important packages, not only the main software, but also the add-on packages that are not installed by default.*

4.2 Results

Reproduced results. The reproducibility run recreated Figures 2–4 in Section 5.1 Synthetic Pipelines, showing results similar to the figures in the paper. This confirms the main finding in the paper. That is, the proposed methods in the paper outperform state-of-the-art methods in precision, recall, and F-measure (for synthetic datasets).

Results not produced. The reproducibility run did not generate figures 5–6 in Section 5.2 Scalability, or Figure 7 in Section 5.3 Real-World Pipelines.

A REPRODUCIBILITY SUBMISSION

```
=========================================================================

Readme for reproducibility submission of paper ID mod0617
ACM Digital library: https://dl.acm.org/doi/10.1145/3318464.3389763

Please follow this: https://github.com/VIDA-NYU/BugDoc/tree/sigmod20#2-experiments

A) Source code info
Repository: https://github.com/VIDA-NYU/BugDoc/tree/sigmod20
Programming Language: Python, Java
Packages/Libraries Needed: Java 1.8.0_144, Python 3.6, PostgreSQL 11
Database instance: There should be a database instance with the following configuration
(https://github.com/VIDA-NYU/BugDoc/blob/sigmod20/sigmod20/experiments/db.conf)

B) Datasets info
Datasets will be generated by the experiments.

C) Hardware Info
Any Unix machine with the libraries above installed could be used. the experiments were
executed on a MacBook Pro, running macOS Catalina version 10.15.7.
C1) Processor: 2.2 GHz Quad-Core Intel Core i7
C2) Memory 16 GB 1600 MHz DDR3
C3) Secondary Storage: HDD, 20 GB available

D) Experimentation Info
D1) All the files related to the experiments are located under sigmod20/
(https://github.com/VIDA-NYU/BugDoc/blob/sigmod20/sigmod20)
D2) prepareSoftware.sh Install the necessary python libraries
(https://github.com/VIDA-NYU/BugDoc/blob/sigmod20/sigmod20/prepareSoftware.sh)
D3) runExperiments.sh runs all the experiments and plots
(https://github.com/VIDA-NYU/BugDoc/blob/sigmod20/sigmod20/runExperiments.sh)

Please note that there is a specific branch of the repository for the experiments, and
make sure to clone it accordingly:
"git clone --branch sigmod20 git@github.com:VIDA-NYU/BugDoc.git"

=========================================================================
```