Schema Extraction of JSON Reproduction

Ilnaz Tayebi
tayebi01@ads.uni-passau.de

ABSTRACT

This paper presents a report on the project of reproducing the Schema Extraction of JSON, an approach for extracting schemas from JSON and Extended JSON document collections.Due to the fact that the project used the same source code [1] and dataset as the original project named “An Approach for Schema Extraction of JSON and Extended JSON Document Collections” [2], it considers as a reproducible effort. The report summarizes the key results of the reproduction project, and evacuation average processing time and discusses the challenges encountered during the reproduction effort. The importance of using a docker file and GitHub repository in creating a reproducible package is also explained. The paper concludes by discussing the differences between Repeat, Reproducing, and Replicating and why the project is considered as reproduction.

ACM Reference Format:
Ilnaz Tayebi. 2023. Schema Extraction of JSON Reproduction. In *Proceedings of* .
, 2 pages. https://doi.org/10.5281/zenodo.7608177

1 INTRODUCTION

In computer science, reproducibility engineering is a crucial aspect that ensures research results can be verified and reused by other scientists. The purpose of this report is to detail a project I completed as part of the Reproducibility Engineering course. My aim was to create a reproducible package for the project “An Approach for Schema Extraction of JSON and Extended JSON Document Collections” [2]. This report describes the reproduction project details, summarizes the key results, encounters challenges in the reproduction effort, differentiates between repeat, reproduce, and replicate, and justifies why my project focuses on reproduction only.

2 PROJECT DESCRIPTION

The project I worked on involved creating a reproducible package for an existing project called "Json Schema Discovery." This project provides a solution for extracting a schema from a collection of JSON or Extended JSON documents stored in a MongoDB NoSQL document-oriented database. The authors of the paper tackle the challenge of managing large amounts of data without a clear data schema and present a method that aggregates the data to create a schema for each unique structure in the collection. The final outcome is a global schema in JSON Schema format, created by combining these individual schemas into a hierarchical data structure. The project was built using Angular and the results of experiments on real datasets such as DBPedia and Foursquare showed that the generated schemas were equal to or better than those from related work.

The project has a GitHub repository and an accompanying article. According to the article, three experiments were conducted to assess the quality of the generated schemas by JSON Schema Discovery. These experiments were the Quality of JSON Document Mapping for JSON Schema, Processing Time Evaluation, and Comparison with Related Work. The experiments were performed on three different datasets that are not readily available. Additionally, when I attempted to reproduce the project, no license had been assigned to it.

, ,
. https://doi.org/10.5281/zenodo.7608177

3 REPRODUCTION EFFORT

The reproduction project involves dockerizing the existing project and automating all its steps. This project is referred to as a reproduction, rather than repetition or replication because it uses the same source code and data as the main project. Docker is used to package the project as it provides a consistent and reproducible environment for the application to run. Packaging the project in a Docker container makes it possible to distribute the application as a single unit, making deployment, scaling, and management easier. Additionally, using containers offers benefits such as improved stability and reproducibility over time by isolating the application from other applications and the host system.

The first step of the project was to create a docker file for the client side of the project, followed by enabling the API to connect to MongoDB. This required the use of docker-compose, which allows running multiple services simultaneously.

Of the three evaluations mentioned in the article, Processing Time Evaluation has the highest reproducibility. The evaluation of JSON Schema Discovery based on the "Quality of JSON document mapping for JSON Schema" does not provide sufficient detail about the 100% accuracy comparison between the five documents. Additionally, the evaluation of the "Quality of JSON Schema Discovery based on comparison with related work" is not preferable as the source code for the related work is difficult to obtain. As a result, the artifacts for quality evaluation based on Processing Time Evaluation have the highest ability to reproduce. The main difference in our experiment for this evaluation is the difference in operating system and hardware. The original work evaluated the processing time on an Amazon EC2t2.micro instance which is not accessible in our case. The datasets used in the experiment are tracked data from tweets, check-ins, and venues from Foursquare. The results are presented in Table 1, showing the average processing time for the schema extraction process for the datasets.

The JSON Schema Discovery process automation has three main steps: inserting datasets into MongoDB, providing a script to run all schema generation steps, and refining the result for presentation in the report. The insertion of datasets into MongoDB is facilitated by an extra service named mongoseed and the mongorestore command. A shell script is provided to simulate all three schema discovery steps, creating a new account and performing an authentication process to generate a valid token. Due to the large size of the JSON data files, scripts have been implemented to allow users to run the

Collection	N-JSON	RS	TB	TT	T B/TT
$venues	2	257	17	74	99.33
$checkins	11	2	2	35	99.29
$tweets	17	23	16	53	99.38

Table 1: RESULTS FOR FOURSQUARE DATASETS (N-JSON: Number of JSON documents.RS:Raw schemas. ROrd:Raw schemas with ordered structure. TB:Time to obtain the raw schemas per minitue.TT:Total time per min.)

process for all or one of the files. Finally, a small python code is written to generate the final result in the report.

3.1 Challenges Encountered

During my efforts to reproduce the project, I encountered several challenges. Firstly, the documentation was not well-prepared, making it difficult to find a compatible version of the dependencies. The Readme file of the project repositories lists the dependencies but doesn't specify their versions. Although I had access to the Package.json file, the project wouldn't build based on the dependencies listed there. To find the best match for all dependencies, I reviewed the git commits of the original project, which led me to stick with Node version 6.11.2. Furthermore, the version of "rotating-file-stream" in the package.json was incompatible with the rest of the packages. As a solution, I created a patch file to rewrite the "rotating-file-stream" version.

The second major challenge was locating the dataset used in the original project. Since reproducing the project required a new team, but the same experiments, it took a significant amount of time to find the same datasets used by the authors. The project's article referenced three different datasets, but none of them were available with the given references. To solve this issue, we reached out to the author of the article via email and requested the dataset. After a few weeks, the author sent us the dataset. Besides, the project's license was not yet granted and I had to follow up with the authors for it, which took about a month to receive.

Additionally, the article explains the implementation of the JSON discovery schema algorithm, but there is no documentation on how to actually use the application. Upon creating an account and logging in, the user is presented with several forms without any explanation of the fields. The language of the application is Portuguese, making the forms even more unclear. I had to resort to investigating the source code of the project to understand how it works.

We faced another unexpected challenge while working on the project. We encountered unknown errors and after two days of troubleshooting, we discovered that the repository owner had made some commits three days ago. To fix this issue, we found that checking out a specific commit in the Docker file after cloning the repository was the solution. This situation highlights the importance of carefully selecting the correct version of the project from the start, as you may not be aware if the original team is actively working on it. It's crucial to decide which branch and commit to use for your project before you start.

Despite the challenges in reproducing the chosen project, having broad technical knowledge is crucial and can be challenging in the reproduction process. A good understanding of various programming languages enables the reproduction team to better understand the original project. Knowledge of scripting languages such as R or Python can aid in automating analyses. The team should also be familiar with tools such as Docker or cloud platforms like AWS, which help to build and run applications in a stable environment and allow for scaling. Adopting Literate programming, with its focus on human-readable documentation, can improve the accessibility, maintainability, and reproducibility of the code. Additionally, using LaTeX as a typesetting system enhances the ability to produce well-formatted documents.

4 CONCLUSION

In conclusion, reproducibility is a vital aspect of the computer science and research field. It guarantees that the outcomes and experiments of a project can be repeated and verified by others. This endeavor aimed to establish a reproducible package for a JSON Schema Discovery project, facing various obstacles such as locating compatible package versions and the dataset utilized in the project. Docker played a role in dockerizing the project in a persistent environment. Uploading the project on Zenodo archives the reproducible package and ensures its longevity. By creating a reproducible package, I have ensured that the results and methodology of the project can be validated and replicated by others in the future.

This project highlights the significance of reproducibility in computer science and research, giving us a clearer understanding of the difficulties and advantages of creating a reproducible package.

REFERENCES

[1] Sam Anzaroot, Alexandre Passos, David Belanger, and Andrew McCallum. 2017. JSONSchemaDiscovery. https://github.com/gbd-ufsc/JSONSchemaDiscovery.git.

[2] Angelo Augusto Frozza, Ronaldo dos Santos Mello, and Felipe de Souza da Costa. 2018. An Approach for Schema Extraction of JSON and Extended JSON Document Collections. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. 356–363. https://doi.org/10.1109/IRI.2018.00060