Effective Generation of Synthetic JSON data using Large Language Models

1 Context and motivation

Automatic generation of synthetic data is essential in many contexts: for testing data pipelines [8], for privacy preserving purposes [9], for training ML models [5, 10] and for benchmarking NoSQL systems [3, 7]. In many situations, data is represented in JSON and its format specified in JSON Schema, a logical language for describing complex JSON documents. While many practical solutions for generating data from JSON Schema exist [1, 2, 6], they are not satisfactory because their generated data is random, not always valid w.r.t. the input schema and it is not realistic enough to reflect the domain intended by the user specifying the schema.

Effectively generating JSON data from a JSON Schema specification is challenging since it needs to i) deal with infinitely many possible schemas, ii) take several criteria into consideration about how to handle each single schema constraint, iii) adhere to some probabilistic distribution so that the generated examples are realistic enough, and iv) exploit already available data in order to generate new data with some resemblance with the pre-existing data.

Large Language Models (LLMs) proved very effective in generating realistic data [4] but also for taking user prompts into consideration [11]. Our goal is to study how to best exploit LLMs for solving the above challenges and to identify their limitations in terms of generating correct examples. To overcome the situation where the generated data is not correct, we also plan to study automatic data repairing based on LLMs. The idea is to build effective prompts guiding the model in its quest to generate a correct instance.

2 Objectives

The aim of the project is to address the challenges above outlined, by devising a novel approach allowing for specifying preferences to guide the generation process or a description of a probability distribution that should be reflected in the generated samples. More specifically the goal is to:

- study solutions for synthetic data generation in general and for JSON Schema in particular and characterize their abilities and limitations.
- devise and implement static analysis techniques for JSON Schema enabling the generation of prompts that can be effectively used to leverage LLMs in order to generate schema instances
- devise new operators for capturing data distribution in JSON Schema and define a generation process guided by these operators
- identify relevant use cases to assess the effectiveness of the developed approach
- deliver an efficient implementation and conduct an experimental study while disseminating the software product as a library

3 Prerequisite

The applicants should have a background in any of the following fields: data management, programming language, machine learning/statistical modeling. The PhD project is part of a larger research project studying many aspects of JSON Schema [1] and involving researchers from Italy and from Germany following an

[1] `https://github.com/miniHive/JSONAlgebra`

intensive interaction. Fluency in English is, thus, required.

4 Contact information

The PhD project will be held under the joint supervision of Pr. Dario Colazzo (`dario.colazzo@lamsade.dauphine.fr`) and Dr. Mohamed-Amine Baazizi (`mohamed-amine.baazizi@lip6.fr`) following an extensive interaction between two highly ranked research institutions, namely, the LAMSADE lab at PSL University, and the LIP6 lab at Sorbonne University.

References

[1] Lyes Attouche, Mohamed-Amine Baazizi, Dario Colazzo, Giorgio Ghelli, Carlo Sartiani, and Stefanie Scherzinger. Witness generation for JSON schema. *Proc. VLDB Endow.*, 2023.

[2] Clara Benac Earle, Lars-Åke Fredlund, Ángel Herranz, and Julio Mariño. Jsongen: a quickcheck based library for testing json web services. In *Proceedings of the Thirteenth ACM SIGPLAN workshop on Erlang*, pages 33–41, 2014.

[3] Angela Bonifati, Irena Holubová, Arnau Prat-Pérez, and Sherif Sakr. Graph generators: State of the art and open challenges. *ACM Comput. Surv.*, 53(2):36:1–36:30, 2021.

[4] Vadim Borisov, Kathrin SeÃ§ler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language Models are Realistic Tabular Data Generators, April 2023. arXiv:2210.06280 [cs].

[5] Eric Breck, Marty Zinkevich, Neoklis Polyzotis, Steven Whang, and Sudip Roy. Data validation for machine learning. In *Proceedings of SysML*, 2019.

[6] Hugo André Coelho Cardoso and José Carlos Ramalho. Synthetic data generation from json schemas. In *11th Symposium on Languages, Applications and Technologies (SLATE 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[7] Kim et al. M2bench: A database benchmark for multi-model analytic workloads. In *VLDB, to appear*, 2023.

[8] Andrew Habib, Avraham Shinnar, Martin Hirzel, and Michael Pradel. Finding data compatibility bugs with JSON subschema checking. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 620–632, Virtual Denmark, July 2021. ACM.

[9] Zhiqi Huang, Ryan McKenna, George Bissias, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Psyndb: accurate and accessible private data generation. *Proceedings of the VLDB Endowment*, 12(12):1918–1921, 2019.

[10] Ciprian Paduraru and Marius-Constantin Melemciuc. An automatic test data generation tool using machine learning. In *ICSOFT*, pages 506–515, 2018.

[11] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT, February 2023. arXiv:2302.11382 [cs].