Boilerplate code data smell impact on LLMs for Code Generation Tasks

Data hound: analysing smells in large code datasets

Supervisors:
Jonathan Katzy[1]
Răzvan Popescu[1]

Author:
Stefan Minkov

Responsible Professors:
Maliheh Izadi[1]
Arie van Deursen[1]

[1]Delft University of Technology

Introduction

Context

- Data smells are indicators for latent data quality issues. [1]
- There have been reported 71 categories of Data Smells for Coding Tasks[2]. One of them is Boilerplate code, see Figure 1.
- Data Smells are reported to potentially impact the performance of Large Language Models (LLMs) when present in their input data.

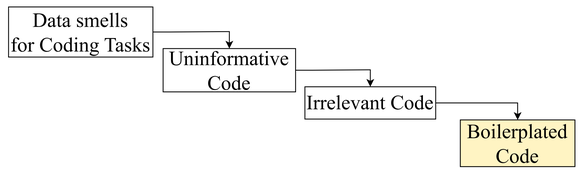


Figure 1. Boilerplate code within the Taxonomy of Data Smells for Coding Tasks

Gap and Motivation

- There is no academic literature quantitatively describing how boilerplate code, as a data smell, impacts code generation by LLMs[2].
- If the data smell significantly impacts LLM performance on coding tasks, it could introduce an unknown bias in their evaluation.

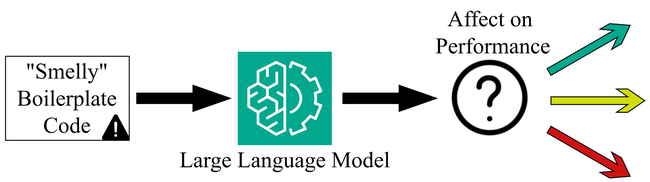


Figure 2. Boilerplate code within the Taxonomy of Data Smells for Coding Tasks

Research Questions

We define the following research questions to address the gap:

- RQ1: How widespread is API usage pattern Boilerplate Code across The Heap?
- RQ2: How does Boilerplate Code affect the code completion performance of an LLM when present in the context window or the target of an inference?
- RQ3: Is Boilerplate Code memorized by LLMs?

Methodology

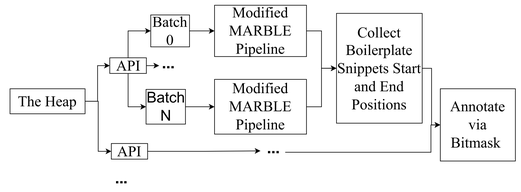


Figure 3. Detection and annotation pipeline

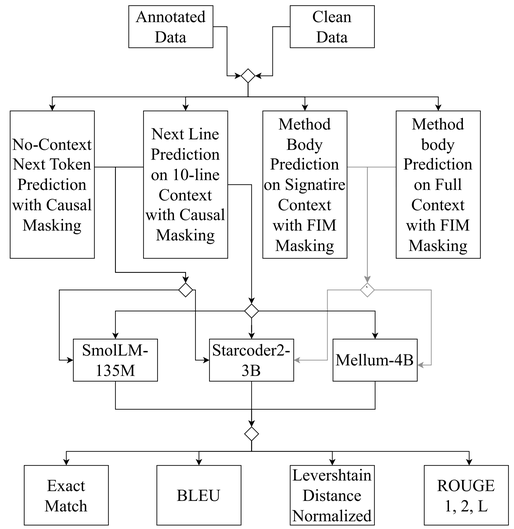
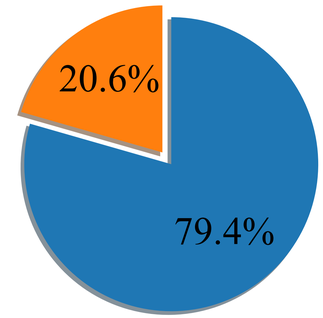


Figure 4. Inferencing and Evaluation pipeline

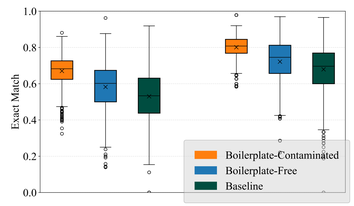
- We leverage a modified version of the MARBLE tool[3] to mine API boilerplate code snippets, as seen in Figure 3
- We develop four experiments in order to evaluate how boilerplate code affects LLM code prediction capabilities when present in the context or target, as seen in Figure 4.
- We developed an additional k-extractibility experiment to measure how much LLMs actually memorize boilerplate code.

Results

- Only 0.3% of the files within The Heap contain the data smell but that accounts to 20,6% of all files containing the 8 target APIs which we investigated.
- LLMs predict code containing boilerplate API usage patterns up to 33 percent points better than.
- Up to 15,8% of boilerplate code is partially memorized by LLMs.



(a) Data Smell part within API investigated files



(b) Representative results for Next-Token_prediction on SmolLM-135M and Starcoder2-3B

Conclusions

- Boilerplate code data smell introduces significant bias in LLM evaluation.
- While the memorization of boilerplate code by LLMs might boost their performance in coding tasks, it may entail legal and privacy consequences for developers that use them.

References

[1] H. Foidl, M. Felderer, and R. Ramler, " Data Smells: Categories, Causes and Consequences, and Detection of Suspicious Data in AI-based Systems ," in *2022 IEEE/ACM 1st International Conference on AI Engineering – Software Engineering for AI (CAIN)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2022, pp. 229–239. [Online]. Available: https://doi.ieeecomputersociety.org/10.1145/3522664.3528590

[2] A. Vitale, R. Oliveto, and S. Scalabrino, "A catalog of data smells for coding tasks," *ACM Trans. Softw. Eng. Methodol.*, vol. 34, no. 4, Apr. 2025. [Online]. Available: https://doi.org/10.1145/3707457

[3] D. Nam, A. Horvath, A. Macvean, B. Myers, and B. Vasilescu, "Marble: Mining for boilerplate code to identify api usability problems," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2019, pp. 615–627.

s.a.minkov-1@student.tudelft.nl