LLMs for Enterprise Data Engineering

Jan-Micha Bodensohn[1,2][0000−0003−4884−0300], Liane Vogel[2][0000−0001−9768−8873], Anupam Sanghi[2][0000−0003−4764−3583], and Carsten Binnig[2,1][0000−0002−2744−7836]

[1] DFKI, Landwehrstraße 50A, 64293 Darmstadt, Germany
[2] Technical University of Darmstadt, Hochschulstraße 10, 64289 Darmstadt, Germany
jan-micha.bodensohn@cs.tu-darmstadt.de

Abstract

Data engineering on tabular data is crucial for transforming raw data sources into a suitable form for downstream tasks like machine learning and analytical query processing. Since handling raw data often entails high manual overheads, automating data engineering tasks like entity matching [5] and column type annotation [3] has long been an active area of research. Recent work shows that Large Language Models (LLMs) achieve state-of-the-art performance on various public benchmarks [4], providing enterprises a promising avenue to automate processes without needing expensive, specialized solutions.

Yet the enterprise domain comes with unique challenges that remain overlooked by existing studies using LLMs, like table sizes and domain knowledge. To address this gap, we analyze enterprise-specific challenges related to data, tasks, and background knowledge, as well as the costs of using LLMs on tabular data. To understand how the differences between scientific benchmarks and enterprise scenarios affect LLM performance, we apply recent LLMs to various enterprise-related downstream tasks on representative customer data.

Data. Existing LLM research primarily uses evaluation datasets based on tables from web sources like Wikipedia and GitHub. Real-world enterprise databases, however, contain tables with vastly different characteristics from those in public benchmarks [2]. For example, our analysis of representative customer data from SAP shows that enterprise tables are substantially larger, containing hundreds of columns and millions of rows. Moreover, they typically represent business objects that span multiple connected tables in non-descriptive schemas. They also often have higher sparsities and contain many non-expressive values such as identifiers and symbolic codes.

To understand how these differences affect LLM performance, we conduct a case study on the task of column type annotation [3]. We find that while recent models perform well on public benchmarks (up to 0.98 F1), they still display severe limitations on our representative customer data (only up to 0.17 F1). Moreover, we show how individual characteristics like wider tables and higher sparsities lead to worse results.

Tasks. In addition to the data challenges, the tasks in enterprise data engineering scenarios are often more complex than their typical formulations in the scientific community. In a case study on the matching of incoming payments to open invoices, an instance of the entity matching problem, we find that state-of-the-art models perform well (up to 0.98 F1) on data comparable to public datasets. However, the addition of enterprise-specific challenges like multi-match scenarios (one payment pays multiple invoices) and multi-table contexts leads to drastic performance drops to 0.66 F1 and 0.58 F1 respectively [1].

Moreover, real-world enterprise scenarios are typically not isolated problems like entity matching, but rather compound pipelines of multiple tasks where errors propagate quickly, posing additional challenges for the automation with LLMs. In a second case study, we use LLMs to integrate one company's customer database into another company's database and show how errors compound across the tasks of schema matching, entity matching, and data transformation.

Knowledge. Data engineering in enterprise scenarios often depends heavily on domain-specific knowledge. One example is the translation of natural language into product-specific query languages like SAP SIGNAL, which is made particularly challenging by the lack of parametric knowledge about the query language. For example, we observe that the generated SIGNAL queries often contain syntax errors. A second challenge lies in customer-specific customizations of applications and data architectures. In contrast to standardized architectures like the SAP schema, these changes are usually not documented publicly. As a result, we observe a bias of the LLMs' parametric knowledge towards the parts of database schemas that are more prominent in public documentation.

Costs. Finally, we discuss the costs of using LLMs for enterprise data engineering. One aspect that drives costs is the immense scale of enterprise databases. Processing a 1TB database with a typical state-of-the-art LLM costs around 1.2M USD. Moreover, the cost per byte for tabular data in TSV format is around twice that of natural language text due to differences in tokenizer fertility. How data engineering problems are typically approached with LLMs can also lead to prohibitive costs. For example, entity matching is usually formulated as comparing pairs of rows, leading to a complexity that grows with the product of the lengths of both tables and is, therefore, not tractable for enterprise use cases.

We want to draw attention to the fact that data engineering in enterprises is often more challenging than portrayed in existing LLM research. We also invite enterprises to share their experience—if not their data—with the research community to ensure that scientific benchmarks reflect real-world problems. Based on our findings, we point towards promising directions to adapt LLMs for enterprise data engineering and support their adoption in industry. Besides standard LLM techniques like fine-tuning and prompt engineering, this includes reformulating established tasks with the particularities of LLMs in mind as well as combining LLMs with existing approaches from the systems community.

Acknowledgments. This work has been supported by the BMBF and the state of Hesse as part of the NHR Program and the HMWK cluster project 3AI. It was also partially funded by the LOEWE Spitzenprofessur of the state of Hesse. We also thank DFKI Darmstadt and hessian.AI for their support.

References

1. Bodensohn, J.M., Brackmann, U., Vogel, L., Sanghi, A., Binnig, C.: Automating enterprise data engineering with LLMs. In: NeurIPS 2024 Third Table Representation Learning Workshop (2024), https://openreview.net/forum?id=m85fYEJtDc
2. Bodensohn, J.M., Brackmann, U., Vogel, L., Urban, M., Sanghi, A., Binnig, C.: LLMs for Data Engineering on Enterprise Data. In: Proceedings of Workshops at the 50th International Conference on Very Large Data Bases, VLDB 2024, Guangzhou, China, August 26-30, 2024. VLDB.org (2024), https://vldb.org/workshops/2024/proceedings/TaDA/TaDA.4.pdf
3. Korini, K., Bizer, C.: Column Type Annotation using ChatGPT (Jul 2023). https://doi.org/10.48550/arXiv.2306.00745, http://arxiv.org/abs/2306.00745, arXiv:2306.00745 [cs]
4. Narayan, A., Chami, I., Orr, L., Ré, C.: Can Foundation Models Wrangle Your Data? Proceedings of the VLDB Endowment **16**(4), 738–746 (Dec 2022). https://doi.org/10.14778/3574245.3574258, https://dl.acm.org/doi/10.14778/3574245.3574258
5. Peeters, R., Bizer, C.: Using ChatGPT for Entity Matching. In: Abelló, A., Vassiliadis, P., Romero, O., Wrembel, R., Bugiotti, F., Gamper, J., Vargas Solar, G., Zumpano, E. (eds.) New Trends in Database and Information Systems. pp. 221–230. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-42941-5_20