Analyzing SQL Query Logs using Multi-Relational Graphs

Andreas M. Wahl and Richard Lenz

Computer Science 6 (Data Management), FAU Erlangen-Nürnberg
{andreas.wahl|richard.lenz}@fau.de

Analytical SQL queries are a valuable source of information. They contain expert knowledge that cannot be inferred from schemas or content alone. Consider, for example, *data lake* scenarios, where relational and semi-structured data sources are combined in a single storage and processing environment. Data lakes often lack a structured curation process [1]. Neither global schemas nor vocabularies might be established and data sources might be disparate. SQL provides effective mechanisms to apply these curation steps during querying in a demand-driven way (e.g. by using aliases, joins, casts, user-defined functions, conditional expressions). Hence, the resulting SQL query logs constitute a dynamic documentation of the data lake and the knowledge gathered by its users through previous pay-as-you-go integration tasks. This knowledge includes the purpose of data sources, their semantics, vocabularies, associations with other data sources, and their temporal and social usage context.

To leverage this knowledge, we have developed an extensible framework for analyzing SQL query logs. Query logs are mapped to a multi-relational [3] graph model. We store query texts and corresponding abstract syntax trees to enable meta-querying for *syntactic features*. However, as SQL allows expressing queries with many different language constructs and the use of aliases, wildcards and unqualified attributes, meta-querying for *semantic features* requires a different query representation. We convert each query to a corresponding relational algebra tree and normalize it using algebraic transformation rules. Each tree is interlinked with a schema lineage tree, which captures attribute lineage and output schemas of each relational operator. Metadata about users, physical time and logical order allows to inspect the social and temporal context of each query. Meta-queries are specified using domain-specific graph traversal expressions.

Our framework can be used for a broad range of application scenarios. It facilitates collaborative data science by locating relevant queries. Other use cases include maintenance and monitoring tasks, schema evolution mechanisms and existing log mining algorithms. We rely on Apache TinkerPop [2] to abstract from vendor-specific graph implementations. TinkerPop enables both interactive meta-querying and complex distributed computations on our graph model.

References

1. Heudecker, N., White, A.: The data lake fallacy: All water and little substance. Gartner Inc. (2014)
2. Rodriguez, M.A.: The gremlin graph traversal machine and language (invited talk). In: DBPL'15 (2015)
3. Rodriguez, M.A., Neubauer, P.: A path algebra for multi-relational graphs. In: ICDEW'11 (2011)

In: M. Leyer (Ed.): Proceedings of the LWDA 2017 Workshops: KDML, FGWM, IR, and FGDB.
Rostock, Germany, 11.-13. September 2017, published at http://ceur-ws.org