supper&supper

Verifizierung und Optimierung einer Graph Datenbank

Projektziel

Graph Datenbanken ermöglichen intuitivere und effizientere Möglichkeiten der Datenabfrage und -analyse als relationale Datenbanken. Aus diesem Grund werden komplexe SQL-Datenbanken häufig in eine nachvollziehbare Graphstruktur migriert. Vor diesem Projekt wurde ein Wissensgraph über Feldversuche erstellt, um Messwerte und Metadaten über globale Feldversuche zu erfassen.[1]

Der Schwerpunkt des Projekts bestand darin, die Graph-Daten für das Training von Machine Learning Modellen zu nutzen. Aus diesem Grund wurden verschiedene Szenarien für Datenexport, -import und automatisierte Verarbeitungspipelines erstellt. In diesem Prozess wurde das gesamte Graph-Schema und die Inhalte gegen die ursprüngliche SQL-Datenbank verglichen und validiert, um zusätzliche Optimierungsansätze abzuleiten. Auch verschiedene Optionen zur Erweiterung des Graph-Schemas mit zusätzlichen Daten für weitere Analysen wurden evaluiert. Insbesondere die Integration von Wetterdaten, die in der SQL-Datenbank nicht abgedeckt wurden, war ein Ziel für dieses Projekt. Es sollte dabei eine effiziente Möglichkeit geschaffen werden, Daten aus dem Graphen zu extrahieren und über Python- und R-Entwicklungsumgebungen zurückzuschreiben. Somit sollen Machine Learning Modelle dynamisch trainiert und auf die Daten des Graphen angewendet werden.

Der neo4j-Graph wird von einer SQL-Datenbank gespeist und sollte für flexible Abfragen verwendet werden, um Daten an Entwicklungsumgebungen weiterzuleiten.

Herausforderungen

Eine verallgemeinerte Dokumentation der komplexen Graphenstruktur ist eine Herausforderung. Außerdem mussten Standardabfragen und Aliase für die Entitäten definiert und dokumentiert werden.

Der Aufbau eines Graph-Schemas bietet eine Menge komplexer Optionen und kann auf sehr unterschiedliche Weise erfolgen. Um das bestehende Graph-Schema zu validieren, mussten zunächst verschiedene Anwendungsfälle für Machine Learning definiert werden. Erst so konnten spezifische Datenbank-Abfragen für den Graphen erstellt und auf Machbarkeit getestet werden. Nicht alle zukünftigen Szenarien waren zu Beginn bekannt. Aus diesem Grund mussten die Test-Szenarien generisch und heterogen sein. Für den Erfolg des Projekts war es auch notwendig, den Graphen auf zuverlässige Weise mit Entwicklungsumgebungen zu kombinieren und zu verbinden.

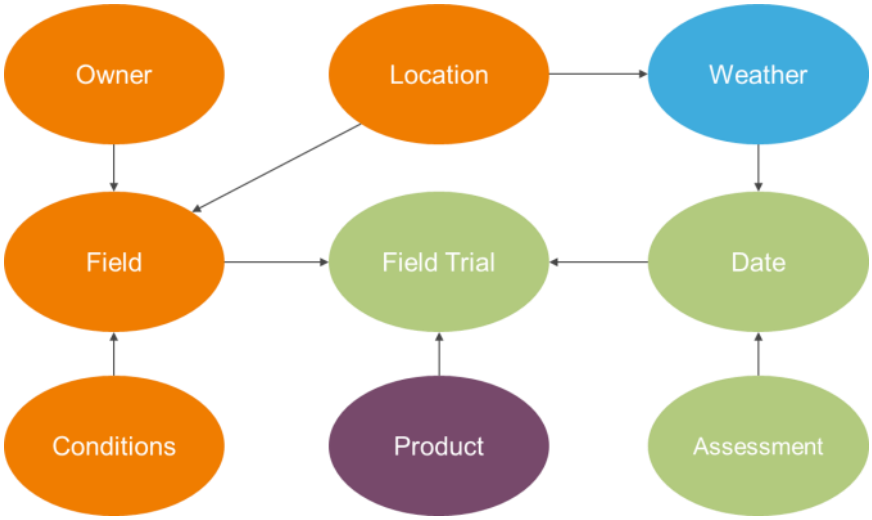
Umsetzung

Verifizierung

Durch den Export von Daten aus der SQL- und der Graph-Datenbank in dasselbe tabellenbasierte Datenformat konnte ein vollständiger Vergleich beider Datenbanken durchgeführt werden. Die Unterschiede wurden bis zu ihrem Kern zurückverfolgt, und es wurden Empfehlungen für Änderungen im Graph-Schema abgeleitet, um sicherzustellen, dass in Zukunft keine Informationen verloren gehen. Außerdem wurden Strategien für die Integration von Wetterdaten und weiteren Datenquellen definiert. Dazu wurden verschiedene Designoptionen für das Schema getestet und für die praktische Anwendung bewertet.

Abfragen und Integration

Um die Datenextraktion zu vereinheitlichen, wurden Richtlinien für Abfragen zur optimalen Datenextraktion sowie Abfragevorlagen und Standard-Aliase definiert. Als Schnittstelle zu externen Entwicklungsumgebungen wurde ein Python-BOLT-Treiber in einer Pipeline für maschinelles Lernen implementiert. Dies ermöglicht einen dynamischen Zugriff auf die Graph Datenbank für verschiedene Analyse- und Machine Learning Anwendungen.



Abstrahiertes Graph-Schema

Projektergebnis

In diesem Projekt wurde unter anderem ein Empfehlungskatalog erstellt:

- Änderungen der Syntax und von Datentypen
- Schema-Abgleiche zur Optimierung häufiger Abfragen zur Leistungssteigerung
- Datenbankoptimierungen
- Korrekturen möglicher Schema-Deadlocks
- Integration von Wetterdaten und weiteren Datenquellen

Als Schnittstelle zu Entwicklungsumgebungen wurde eine Python / R-Pipeline aufgebaut. Durch Integration in Analysewerkzeuge kann der Graph bei Bedarf abgefragt werden, so dass die Ergebnisse für das Training von Machine Learning Modellen genutzt und direkt auf weitere Daten angewendet werden können.

Referenz:

[1] Daniel Burkow, Jens Hollunder, Julian Heinrich, Fuad Abdallah, Miguel Rojas-Macias, Cord Wiljes, Philipp Cimiano and Philipp Senger
A Blueprint for Semantically Lifting Field Trial Data: Enabling Exploration using Knowledge Graphs, December 2019.