

Bachelorarbeit

User-aided Pattern Search and Analysis on Business Graphs

Nutzergesteuerte Graphanalyse und Mustersuche auf
Unternehmensgraphen

Milan Gruner

`milangruner@gmail.com`

Eingereicht am <TBD>

Fachgebiet Informationssysteme

Betreuung: Prof. Dr. Felix Naumann, Michael Loster, Toni Gruetze

Abstract

Costructing a graph made up of thousands of businesses maybe hard, but actually making sense of it is a lot harder. With huge amounts of data being integrated into the data lake every day, automatic methods for finding interesting spots in the graph are needed. This paper discusses different approaches that can be taken to extract useful knowledge from such a graph.

Contents

1	Introduction	4
1.1	Glossary	4
1.2	Motivation	4
1.3	Understanding risk analysis on graphs	4
1.4	Used techniques and related works	4
2	Data structures for business entities	4
2.1	Graph encoding for column family storage	4
2.2	The <i>subject</i> data structure	4
2.3	A versioning scheme that stands the test of time	4
3	Architecture	4
3.1	Modularizing Spark jobs	4
3.2	Data flow using column family storage	4
3.3	Using Apache Spark and Cassandra for Graph Analysis	4
3.4	Writing efficient Spark code for Graphs	4
3.5	Cassandra-specific optimizations	4
4	Pattern Search	4
4.1	Discerning patterns from randomness	4
4.2	Pattern types and their applications	4
4.3	Operating on graph diffs	4
5	Pattern Analysis	4
5.1	User-aided approaches for Pattern Categorization	4
6	Graph Summarization	4
6.1	What users actually want to see	4
6.2	Compressing graph information to the bare minimum	4
6.3	Presenting graph data appealingly	4
7	Lessons learned	4
7.1	Benchmarks and Experiments	4
7.2	Design decisions and trade-offs	4
7.3	Technical challenges	4
8	Literature	4

1 Introduction

1.1 Glossary

1.2 Motivation

1.3 Understanding risk analysis on graphs

1.4 Used techniques and related works

2 Data structures for business entities

2.1 Graph encoding for column family storage

2.2 The *subject* data structure

2.3 A versioning scheme that stands the test of time

3 Architecture

3.1 Modularizing Spark jobs

3.2 Data flow using column family storage

3.3 Using Apache Spark and Cassandra for Graph Analysis

3.4 Writing efficient Spark code for Graphs

3.5 Cassandra-specific optimizations

4 Pattern Search

4.1 Discerning patterns from randomness

4.2 Pattern types and their applications

4

4.3 Operating on graph diffs

5 Pattern Analysis

5.1 User-aided approaches for Pattern Categorization

References