Duplicate-Aware Federated Query Processing over the Data Web

ABSTRACT

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems—*Distributed databases*

General Terms

Algorithms, Experimentation, Theory

Keywords

Federated queries, SPARQL, deduplication

1. INTRODUCTION (AN+MS)

Over the last years, the Linked Data Web has developed into a large compendium of interlinked data sets from diverse domains. One of the central principles underlying the architecture of these data sets is the reuse of URIs and vocabularies as well as the linking of knowledge bases. One of the results of this architectural choice is that certain queries can only be answered by retrieving information from several knowledge bases. This type of querying, called *federated querying*, is of central importance for manifold applications such as question answering, knowledge retrieval and data integration. In addition to the information necessary to answering queries being distributed, certain pieces of information (i.e., triples) can be found in several knowledge bases. For example, the name of movie directors can be found both in DBpedia and LinkedMDB. Similarly, the authors of papers can be found in both the ACM and DBLP libraries. We call triples that can be found in several knowledge bases across the Web of Data *duplicates*.

While the importance of federated queries over the Web of Data has been stressed in previous work, the impact of duplicates has not received much attention. Yet (as we will show in the remainder of this paper), a duplicate-aware approach to query processing can lead to more time-efficient and effective algorithms for federated queries. In this paper, we address this drawback by presenting a duplicate-aware approach for query processing based on min-wise independent permutation (MIPS) vectors. Our approach is able to predict the amount of new information contained in a knowledge base with a higher accuracy than the state of the art, leading to a better performance with respect to source ranking for both whole queries and fragment of queries as well as better result set size approximation. In the rest of this paper, we aim to show experimentally that our approach outperforms the state-of-the-art by running it against ?? methods on ?? queries. We begin by giving a brief overview of the state of the art in federated query processing. In addition, we argue for the use of MIPS for the approximation of result size sets. Thereafter, we present our duplicate-aware federated query processing approach. After an overview of the datasets used in this paper, we present experimental results which corroborate the superior efficiency of our approach. We conclude the paper with a discussion of our findings and an overview of future work.

2. RELATED WORK (MS)

2.1 Federated SPARQL queries

2.2 Filters

Bloom MIPS etc.

3. NOTATION

In this section, we present the core of the notation that will be used throughout this paper. We denote data sources with S and the total number of data sources with n. The set of all possible result sets is denoted R while the set of all possible SPARQL queries is labeled with Q. A data source ranking function $rank : S \times Q \rightarrow \{1 \dots n\}$ is a function that assigns a ranking to each data source given a particular query $q \in Q$. Note that for any source ranking function $rank$, we assume that $\forall S, S' \forall q \in Q : S \neq S' \iff rank(S, q) \neq rank(S', q)$. A result set estimation function $est : S \times Q \rightarrow \mathbb{N}$ aims at approximating the size of the result set that will be returned by a given query. Note that this function plays a crucial role in the processing of federated queries as it is most commonly used to decide upon the ranking of data sources for a given query. The aim of a federated query system such as the one described in this work is thus to optimize its estimation function est so as to ensure a ranking of the source close to the optimal ranking.

4. APPROACH (MS)

4.1 Overview

4.2 MIPS

Vector Construction Union, overlap

4.3 Index construction

Compression Ratio Setting

4.4 Source Selection

4.5 Source Ranking

Result set estimation

4.6 Subquery generation

5. EXPERIMENTS AND RESULTS (AN)

5.1 Experimental Setup (AN)

5.1.1 Datasets (3 datasets)

5.1.2 Queries (6 types)

5.1.3 Metrics (MSE)

5.2 Results (AN)

5.2.1 Ranking Error

5.2.2 Result Set Estimation Error

5.2.3 Execution Time

5.2.4 Size of MIPS vectors (on largest dataset)

6. DISCUSSION (AN)

7. REFERENCES