Triple Pattern Join Cardinality Estimations over HDT with Enhanced Metadata

Elena Wössner[1], Chang Qin[1], Javier D. Fernández[2,3], and Maribel Acosta[1]

[1] Institute AIFB, Karlsruhe Institute of Technology (KIT)
{elena.woessner|chang.qin}@student.kit.edu, maribel.acosta@kit.edu
[2] Vienna University of Economics and Business
[3] Complexity Science Hub Vienna jfernand@wu.ac.at

Abstract. In this work, we present HDT-STATS, an extension to the HDT operations, to compute further metadata when evaluating triple patterns over RDF graphs represented with HDT. Then, we propose a novel model that relies on the HDT-STATS metadata, as well as the distinct position of SPARQL variables, to estimate the cardinality of joins between triple patterns. Our preliminary results suggest that our approach produces more accurate cardinality estimations than existing solutions.

1 Introduction

HDT (Header, Dictionary, Triples) [2] is a compact serialization format for RDF graphs. HDT supports triple-pattern-based querying and provides metadata about the estimated number of matched RDF triples. Therefore, the execution of SPARQL queries over HDT requires the combination of the results of evaluating individual triple patterns. Current query engines against interfaces over HDT (e.g., nLDE [1]) implement optimization techniques that rely on simple cardinality estimations to devise query plans that reduce execution time. However, the cardinality estimation of join operators with insufficient metadata is rather challenging and, in most cases, may lead to inefficient query plans. To assist query engines in devising more effective query plans, in this work, we propose HDT-STATS, a novel interface that enhances the metadata provided by HDT. Besides cardinality estimations for triple patterns, our approach provides the number of distinct subjects, predicates, and objects in result sets. Then, we propose a novel model for estimating the cardinality of joins of triple patterns, based on the available metadata provided by HDT-STATS. This work covers the estimation of the cardinality of subject-subject, object-object, and subject-object joins. In contrast to state-of-the-art cost models for SPARQL query optimization, e.g., CostFed [3], our approach combines the HDT-STATS metadata in a novel way that produces more accurate estimates. To evaluate our solution, we conducted an empirical study using 287 joins between triple patterns over DBpedia.

2 Our Approach

2.1 Evaluation of Triple Patterns with HDT-STATS

HDT supports rank and select operations [2], which allow for accessing triples in the compressed RDF graph. With these operations, HDT is able to evaluate triple patterns and compute cardinality estimates of the number of triples that belong the solution. In this work, we propose HDT-STATS, an interface to support the retrieval of further metadata about the evaluation of triple patterns over HDT data structures. Formally, the evaluation of a triple pattern over an HDT graph when using the HDT-STATS is defined as follows.[1]

Given a triple pattern tp, an HDT RDF graph G, the solution of evaluating a tp over G with HDT-STATS is a 5-tuple $(\Psi, card, ds, dp, do)$, where:

- Ψ: the result set, i.e., the subset of RDF triples in the graph G that match the triple pattern tp, i.e., $\Psi = \{\mu(tp) \mid dom(\mu) = vars(tp)\ and\ \mu(tp) \in G\}$,
- $card$: estimated value for the total number of solutions $|\Psi|$,
- ds: estimated number of distinct subjects in Ψ, i.e., $|\{s \mid (s,p,o) \in \Psi\}|$,
- dp: estimated number of distinct predicates in Ψ, i.e., $|\{p \mid (s,p,o) \in \Psi\}|$,
- do: estimated number of distinct objects in Ψ, i.e., $|\{o \mid (s,p,o) \in \Psi\}|$.

HDT-STATS is able to compute the metadata (ds, dp, do) efficiently, as it relies on light-weight extensions to the HDT operations. This metadata is then exploited by the the cardinality estimation model for joins between triple patterns.

2.2 Cardinality Estimation Model

One of the key components in cost-based query optimization techniques is the estimation of the number of intermediate results produced by the operators in a query plan. In this work, we focus on the estimation of cardinalities when joining triple patterns, i.e., $\widehat{card}(tp_i \bowtie tp_j)$ with tp_i and tp_j triple patterns.

To estimate the cardinality of joins, we propose a light-weight model that considers the novel metadata retrieved with HDT-STATS, i.e., distinct subjects, distinct predicates, and distinct objects contained in a result set. Considering this metadata, our proposed model distinguishes between the different types of joins that may occur when joining triple patterns. In this work, we focus on subject-subject, object-object, and subject-object joins. In the following, we propose separate estimations for each join type.

Subject-Subject Joins. Subject-subject joins occur when two triple patterns exclusively share a common variable in their subject position. To estimate the cardinality of subject-subject joins, our proposed model considers the number of distinct subjects contained in the solutions of the joined triple patterns. The number of distinct subjects allows for estimating the selectivity of each triple pattern in terms of subjects. In this case, the selectivity of subjects represents the number of times (on average) that an arbitrary subject occurs in the solution

[1] We assume the terminology of SPARQL query evaluation as in the literature.

of a triple pattern. For example, subject selectivity equals to 1 indicates that every subject occurs exactly one time in the solution set. To calculate such a selectivity, we just divide the number of solutions by the number of distinct subjects. Then, to calculate the cardinality of the join, our model divides the total number of possible answers by the maximum number of distinct subjects, which represents an upper bound on the number of subjects that will match. Based on this, we define our proposed model in the following.

Given triple patterns $tp_i = (?x, pi, oi)$ and $tp_j = (?x, pj, oj)$. Assume that $(\Psi_i, card_i, ds_i, dp_i, do_i)$ and $(\Psi_j, card_j, ds_j, dp_j, do_j)$ are the results of evaluating triple patterns tp_i and tp_j over a graph G, respectively. The estimated cardinality of performing $tp_i \bowtie tp_j$, denoted $\widehat{card}(tp_i \bowtie\ tp_j)$, is defined as follows:

$$\widehat{card}(tp_i \bowtie\ tp_j) = \frac{card_i \cdot card_j}{max(ds_i, ds_j)} \tag{1}$$

Object-Object Joins. Object-object joins occur when two triple patterns exclusively share a common variable in their object position, i.e., $tp_i = (si, pi, ?x)$ and $tp_j = (sj, pj, ?x)$. Analogous to the subject-subject estimation, we proposed a model that considers the selectivity of objects in the result sets, as follows:

$$\widehat{card}(tp_i \bowtie\ tp_j) = \frac{card_i \cdot card_j}{max(do_i, do_j)} \tag{2}$$

Subject-Object Joins. Subject-object joins occur when two triple patterns exclusively share a common variable in a subject position in one triple pattern and in object position in the other triple pattern, i.e., $tp_i = (?x, pi, oi)$ and $tp_j = (sj, pj, ?x)$. Similarly to previous cases, our model considers the number of distinct subjects and the number of distinct objects involved in the join. The maximum number of subjects or objects corresponds to an upper bound on the number of resources that match. The proposed estimation is as follows:

$$\widehat{card}(tp_i \bowtie\ tp_j) = \frac{card_i \cdot card_i}{max(ds_i, do_j)} \tag{3}$$

3 Experimental Study

Dataset and Queries. We used the benchmark queries over DBpedia (v.2015) presented in the work by Acosta and Vidal [1]. Per query, we generated all possible pairs of triple patterns that share variables in common. This resulted in 287 triple pattern joins, with the following distributions: 255 subject-subject joins, 23 object-object joins, and 9 subject-object joins.

Approaches. To measure the effectiveness of our proposed cardinality estimation model, we compare our approach with the model computed by CostFed [3].

Metrics. We report on the absolute error (**AE**) and the relative error (**RE**) of the models. AE is the absolute difference between the estimated cardinality and the actual number of answers produced by the join. RE is computed as AE divided by the actual number of answers. In addition, we report on the Pearson correlation coefficient between the estimated and real cardinalities.

Table 1. Effectiveness of the proposed model. Mean absolute error (Mean AE) and mean relative error (Mean RE) of cardinality estimations of different join types: subject-subject (S-S), object-object (O-O), subject-object (S-O). (Lower is better).

Metrics	CostFed	Our Model	CostFed	Our Model	CostFed	Our Model
	S-S	S-S	O-O	O-O	S-O	S-O
Mean AE	1.90E+6	4.82E+4	6.84E+6	4.75E+6	4.28E+4	1.67E+4
Mean RE	14.01	4.54	11.68	1.65	9.34	7.21

3.1 Effectiveness of the Proposed Cardinality Estimations

In this study, we compare the effectiveness of our proposed cardinality estimation with the model computed by CostFed. CostFed is a SPARQL federated engine that implements a cost model based on cardinality estimations, which relies on similar statistics. Therefore, a direct comparison with our techniques is possible.

Table 1 reports on the effectiveness of the studied approaches, measured by absolute and relative errors. The results indicate that, on average, our proposed model produces more accurate estimations for all types of joins than the ones implemented by CostFed. In particular, our approach performs best in predicting the cardinalities of object-object joins followed by subject-subject-joins.

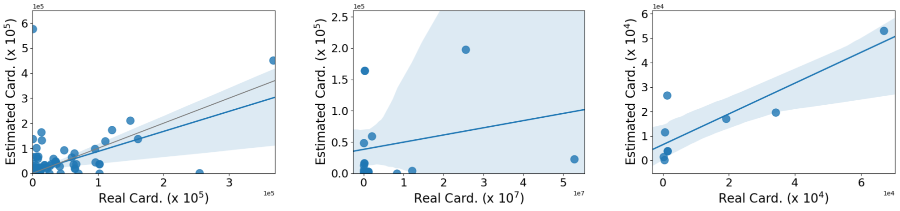
A closer look into the raw results reveals that there is a particular case when both models produce the same estimation: when the selectivity of subjects or objects in a result set is equal to 1. For the other cases, our proposed cost model typically produce estimates smaller than the ones by CostFed. This indicates that even when both models overestimate the cardinality of a join, our prediction is still closer to the real number of answers. This is confirmed by the mean relative errors achieved in all types of joins (cf. Table 1).

Lastly, our study reveals that both CostFed and our approach produce more accurate estimates for the cardinality of object-object joins than for subject-subject joins. This could be a result of the topology of the DBpedia graph, where estimating the number of subjects that match the triple patterns drastically change when using different constants in the patterns. Still, as the number of object-object joins in our evaluation is rather low (23 joins), we need to conduct further experiments to derive concrete conclusions about the behavior of the approaches in the object-object case.

3.2 Analysis per Join Type

In this study, we focus on understanding the behavior of our proposed cardinality estimation model. We compare the estimated cardinality of our approach with the actual number of answers produced by the joins. Figure 1 reports on the actual number of answers vs. the estimated cardinalities by our approach.

For **subject-subject joins**, we first realize that a few data points allows our approach to achieve very high correlation. This is the case when joins produce a large number of results and that our model was able to estimate accordingly. To be able to properly analyze the results on the majority of the triple patterns,



(a) S-S Joins (b) O-O Joins (c) S-O Joins

Fig. 1. Behavior of the proposed model per join type. Correlation between the real cardinality (x-axis) and estimated cardinality with our approach (y-axis).

we focused on the results for joins that produce less than 1M answers (cf. Figure 1(a)). In this case, the Pearson correlation achieved is 0.543. Despite the moderate positive correlation, we observe that our model tends to overestimate the cardinality of subject-subject joins over DBpedia.

Figure 1(b) reports the results of **object-object joins**. Despite achieving the lowest errors (AE and RE) in this type of join, our proposed model is still volatile, i.e., there are no clear patterns of under- or over-estimations. This is confirmed by the Pearson correlation coefficient (0.175).

The results for **subject-object joins** are reported in Figure 1(c). We observe that the model also tends to overestimate the cardinalities. In this case, the correlation achieved is 0.863. Still, these results are considered preliminary as the benchmark did not have sufficient occurrences of this type of join.

4 Conclusions and Future Work

We have presented HDT-STATS to support the retrieval of further metadata during the evaluation of triple patterns over RDF graphs compressed with HDT. Based on this metadata, we propose a novel cardinality estimation model that considers the type of the join between pairs of triple patterns. Our results on over 287 joins indicate that our proposed model outperforms state-of-the-art solutions. For future work, we plan to extend our model to cover further join types and to conduct further experimental studies over other RDF graphs.

References

1. Maribel Acosta and Maria-Esther Vidal. Networks of linked data eddies: An adaptive web query processing engine for rdf data. In *ISWC*, pages 111–127, 2015.
2. Javier D. Fernández, Miguel A. Martínez-Prieto, Claudio Gutiérrez, Axel Polleres, and Mario Arias. Binary RDF representation for publication and exchange (HDT). *J. Web Semant.*, 19:22–41, 2013.
3. Muhammad Saleem, Alexander Potocki, Tommaso Soru, Olaf Hartig, and Axel-Cyrille Ngonga Ngomo. Costfed: Cost-based query optimization for SPARQL endpoint federation. In *SEMANTICS*, pages 163–174, 2018.