Parallel Structural Graph Summarization - Compute k-bisimulation Summaries using the FLUID-based (B,S,R)-Algorithm

Jannik Rau

Mai 2020

1 Motivation

Graphs, for example in the form of a directed, edge-labeled graph, become an increasingly popular data-modeling paradigm [12]. A common example of a directed, edge-labeled graph is that of an RDF [7] graph, which contains information about (real-world) entities - corresponding to the graphs nodes - in the form of relations between them - corresponding to the graphs edges. RDF graphs, also termed as *Knowledge Graphs*, emerged and grew in both, the open-source domain, as well as the enterprise domain [12]. The applications of such graphs are numerous, ranging from search engines, recommendation systems of any kind, fraud detection and business intelligence, to simple information holders.

However, if the graphs size becomes bigger, certain problems arise, like storing, indexing and querying the graph or understanding/visualizing the graph in general [4].

One technique to approach these problems is *Graph Summarization* [6]. Depending on the use case, one can summarize the graph with respect to structural features (e.g. incoming/outgoing paths), statistical measures (e.g. occurrences of specific nodes) or frequent patterns found in the graph [6]. This results in a usually smaller sized *Summary Graph*, which contains an approximation of or exactly the same information as the original graph. Therefore, the summary graph can be used to overcome the aforementioned problems.

2 Related Work

This thesis targets a structural graph summarization technique based on quotients, which summarizes a graph according to an equivalence relation $\sim \subseteq V \times V$ defined on the nodes V of the graph G. The nodes VS in the summary graph GS correspond to the equivalence classes A of the relation $\sim$. Before going into more detail, a brief overview of other existing summarization techniques is given.

Beside of the quotient summaries, the other big class of structural summarization techniques is called non-quotient summaries. In this setting, the summary graph GS consists of *supernodes* vs, which group together nodes v of the original graph G based on properties like data paths [9], or node attributes in general [6]. The main difference to quotient summaries is that in non-quotient summaries a node v can belong to zero or multiple supernodes vs, whereas in quotient summaries every node v has exactly one corresponding summary node vs, which is the equivalence class of v under $\sim$ [6].

Different to structural graph summarization, one can also summarize a graph G according to frequent patterns in G or with respect to statistical properties of G, like frequency of node attributes [6].

As mentioned before, the scope of this thesis is quotient graph summarization. More specifically, the evaluation of a parallel *generic* structural quotient algorithm [2]. The algorithm is termed generic, because it summarizes a graph with respect to a user specified equivalence relation $\sim$, defined in the formal language *FLUID* [3]. Thus, the algorithm is not specific to one graph summary model or task, as it is the case with a lot of existing research in the domain of structural graph summarization [2].

In the existing research, one can observe that *k-bisimulation* is a popular feature for structural graph summarization [3]. A bisimulation is an equivalence relation defined on the nodes of a directed, edge-labeled graph G. Two nodes u,v are considered (backward) forward-bisimilar, if they share the same label and for every (incoming) outgoing neighbour u' connected to u via an edge labeled p, there exists a respective (incoming) outgoing neighbour v' connected to v via an edge labeled p, and vice versa. Furthermore, the two neighbours u' and v' have to be bisimilar as well [13, 19].

It is important to note, that this definition corresponds to a *complete* bisimulation. For the neighbours u' and v' to be bisimilar, their (incoming) outgoing neighbours, if any existing, u'' and v'' have to be bisimilar as well, and so on. A stratified bisimulation, or k-bisimulation relaxes the former

notion of bisimulation, such that only the neighbours of distance k have to be bisimilar as well.

The following works make use of k-bisimulation in their approach to structural graph summarization.

Buneman et al. make use of forward k-bisimulation in the problem of RDF graph alignment [5]. Summarizing the union of two consecutive versions $G_{\text{union}} = G_1 \cup G_2$ of an RDF graph with respect to k-bisimulation, puts nodes to be aligned in the same partition. Additionally to k-bisimulation, they use a similarity measure to further refine the initial k-bisimulation partition, as it doesn't capture all nodes to be aligned. The focus of their work is the optimization of the alignment process, so that every node pair (v_1, v_2), with $v_1 \in G_1$ and $v_2 \in G_2$, which have to be aligned is identified and not the construction of a k-bisimulation-based partition of G. Therefore, this thesis research questions will not consider this work.

Combining forward- and backward-bisimulation, Tran et al. compute a structural index for graphs based on forward-backward k-bisimulation [19]. Moreover, they parameterize their notion of bisimulation to a forward-set L_1 and a backward-set L_2, so that only labels $l \in L_1$ are considered for forward-bisimulation and labels $l \in L_2$ for backward-bisimulation. However, similar to Buneman et al. the particular focus of their work is not the actual construction of the structural index, e. g., the bisimulation partition. Rather, they evaluate how one can efficiently optimize query processing on semi-structured data using such an index graph [19]. As a consequence, this work is not considered either.

Schätzle et al. compute a forward k-bisimulation on RDF graphs in a sequential and a distributed setting [17]. For a small synthetic dataset ($\sim 1M$ RDF-triples) the sequential algorithm slightly outperforms the distributed one, whereas for increasing size of the dataset, the distributed algorithm clearly outperforms the sequential one.

Luo et al. examine structural graph summarization with respect to forward k-bisimulation in an external memory solution [15]. Furthermore, they empirically observe, that for values of $k > 5$, the summary graphs partition blocks do not or just barely change. Therefore they state, that for summarizing a graph with respect to k-bisimulation, it is sufficient to summarize up to a value of $k = 5$ [14].

Different to these approaches, Kaushik et al. propose a summarization technique based on backward k-bisimulation [13]. Their A(k)-Index serves

as a framework for computing backward k-bisimulation of a graph G, which results in an index graph $I(G)$.

3 Research Questions

All of the latter three mentioned summarization techniques are based on k-bisimulation, but implemented in a specifically designed algorithm. Furthermore, as a k-bisimulation can be modeled via the formal language FLUID [3], one can execute the specific summarization model through the generic (B,S,R)-algorithm.

This leads to the research questions of this thesis. How does the generic (B,S,R)-algorithm perform for a specific k-bisimulation-based graph summary model in comparison to the respective, specifically designed algorithm. More concrete, how does the value k influence the performance, how do the dataset characteristics like size and diversity influence the performance. Consequently, the main research question is, if the FLUID-based, generic (B,S,R)-algorithm can be used as a general framework for k-bisimulation-based structural graph summaries.

To examine these questions, (RQ 1) first, the algorithms of all of the chosen k-bisimulation approaches are checked for availability. If an algorithm is not available or outdated in its implementation, it will be reimplemented according to the respective paper.

(RQ 2) Afterwards, the respective summary models SM_i will be executed with increasing values for k (1) through their specific implementation and (2) through the generic (B,S,R)-algorithm.

(RQ 3) In addition to that, the degree of possible parallelism of each algorithm will be examined.

For execution and evaluation, the following datasets and measures have been chosen respectively.

4 Datasets

The use of real-world datasets when evaluating graph summarization algorithms is of great importance, as synthetic datasets lack in capturing the structure of real-world graphs [2,14]. Therefore, four real-world datasets and only two synthetic datasets were chosen for the experiments. Moreover, the

(B,S,R)-algorithm has the capability of incrementally updating a summary graph, if the corresponding original graph has changed, making evolving graphs a possible target for experiments. However, due to time constraints the update functionality will not be examined and thus all of the graphs are considered as static.

4.1 Real-World Datasets

Concerning the real-world datasets, four datasets of different size and diversity were chosen.

First, the *Billion Triple Challenge 2019* (BTC2019) [11] contains around 256M unique RDF triples obtained from more than 2.6 million RDF documents.

Second, the *LOD-a-lot* dataset [8] provides more than 28B unique triples from the *LOD Laundromat*[1] service, which crawls, processes and republishes data from different sources. The provided triples originate from around 650K different datasets.

Third, a rather small dataset, the *Jamendo* dataset [16], contains approximately 1.1M unique triples of various information about musical artists and their published work.

Lastly, the Linked Movie DataBase (LinkedMDB) dataset connects movies, directors and artists with their appropriate relationships [10]. It consists of approximately 6.1M unique triples.

4.2 Synthetic Dataset

For the two synthetic datasets, two benchmarks, originally developed for measuring the performance of storage systems, which act as SPARQL endpoints, were chosen.

The *Berlin SPARQL Benchmark* (BSBM) provides an RDF-Data generator for an e-commerce use case, including products, vendors and consumer reviews [1].

Settled in a different use case, the *SPARQL Performance Benchmark* (SP^{2}Bench) contains an RDF-Data generator for DBLP-like models [18].

Another notable synthetic benchmark dataset is LUBM[2], however it will not be used for experiments.

[1] `https://lodlaundromat.org`

[2] `http://swat.cse.lehigh.edu/projects/lubm/`

5 Measures

The thesis scope is to evaluate the algorithms' performance. Hence, the two main performance indicators are the algorithms run-time and the size of any crucial data structure used during the algorithm, e.g. a dictionary.

Furthermore, the time as well as the size values are measured for every individual, increasing value of k.

6 Schedule

The thesis approximate schedule is as follows:

- Implementation - 4 months
  - Algorithms - 3 months (one month per algorithm)
  - Experimental framework (setup, measures, plots, ...) - 1 month
- Evaluation - 2 months
  - Executing experiments - 4 (algorithms) $\cdot$ 6 (datasets - benchmarks are also considered as only one) = 24 experiments, so approx. 1 month
  - Evaluating experiment results and incorporating them into the thesis - 1 month

References

[1] Christian Bizer and Andreas Schultz. The berlin SPARQL benchmark. *Int. J. Semantic Web Inf. Syst.*, 5(2):1–24, 2009.

[2] Till Blume, David Richerby, and Ansgar Scherp. Incremental and parallel computation of structural graph summaries for evolving graphs. In Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux, editors, *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 75–84. ACM, 2020.

[3] Till Blume, David Richerby, and Ansgar Scherp. FLUID: A common model for semantic structural graph summaries based on equivalence relations. *Theor. Comput. Sci.*, 854:136–158, 2021.

[4] Angela Bonifati, Stefania Dumbrava, and Haridimos Kondylakis. Graph summarization. *CoRR*, abs/2004.14794, 2020.

[5] Peter Buneman and Slawek Staworko. RDF graph alignment with bisimulation. *Proc. VLDB Endow.*, 9(12):1149–1160, 2016.

[6] Sejla Cebiric, François Goasdoué, Haridimos Kondylakis, Dimitris Kotzinos, Ioana Manolescu, Georgia Troullinou, and Mussab Zneika. Summarizing semantic graphs: a survey. *VLDB J.*, 28(3):295–327, 2019.

[7] Richard Cyganiak, David Wood, and Markus Lanthaler. Rdf 1.1 concepts and abstract syntax. `https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/`, 2014. Accessed: 2021-04-16.

[8] Javier D. Fernández, Wouter Beek, Miguel A. Martínez-Prieto, and Mario Arias. Lod-a-lot - A queryable dump of the LOD cloud. In Claudia d'Amato, Miriam Fernández, Valentina A. M. Tamma, Freddy Lécué, Philippe Cudré-Mauroux, Juan F. Sequeda, Christoph Lange, and Jeff Heflin, editors, *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*, volume 10588 of *Lecture Notes in Computer Science*, pages 75–83. Springer, 2017.

[9] Roy Goldman and Jennifer Widom. Dataguides: Enabling query formulation and optimization in semistructured databases. In Matthias Jarke, Michael J. Carey, Klaus R. Dittrich, Frederick H. Lochovsky, Pericles Loucopoulos, and Manfred A. Jeusfeld, editors, *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*, pages 436–445. Morgan Kaufmann, 1997.

[10] Oktie Hassanzadeh and Mariano P. Consens. Linked movie data base. In Christian Bizer, Tom Heath, Tim Berners-Lee, and Kingsley Idehen, editors, *Proceedings of the WWW2009 Workshop on Linked Data on the Web, LDOW 2009, Madrid, Spain, April 20, 2009*, volume 538 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[11] José-Miguel Herrera, Aidan Hogan, and Tobias Käfer. BTC-2019: the 2019 billion triple challenge dataset. In Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtech Svátek, Isabel F. Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon, editors, *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, volume 11779 of *Lecture Notes in Computer Science*, pages 163–180. Springer, 2019.

[12] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *CoRR*, abs/2003.02320, 2020.

[13] Raghav Kaushik, Pradeep Shenoy, Philip Bohannon, and Ehud Gudes. Exploiting local similarity for indexing paths in graph-structured data. In Rakesh Agrawal and Klaus R. Dittrich, editors, *Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, February 26 - March 1, 2002*, pages 129–140. IEEE Computer Society, 2002.

[14] Yongming Luo, George H. L. Fletcher, Jan Hidders, Paul De Bra, and Yuqing Wu. Regularities and dynamics in bisimulation reductions of big graphs. In Peter A. Boncz and Thomas Neumann, editors, *First International Workshop on Graph Data Management Experiences and Systems, GRADES 2013, co-located with SIGMOD/PODS 2013, New York, NY, USA, June 24, 2013*, page 13. CWI/ACM, 2013.

[15] Yongming Luo, George H. L. Fletcher, Jan Hidders, Yuqing Wu, and Paul De Bra. External memory k-bisimulation reduction of big graphs. In Qi He, Arun Iyengar, Wolfgang Nejdl, Jian Pei, and Rajeev Rastogi, editors, *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 919–928. ACM, 2013.

[16] Yves Raimond and Mark B. Sandler. A web of musical information. In Juan Pablo Bello, Elaine Chew, and Douglas Turnbull, editors, *ISMIR*

2008, 9th International Conference on Music Information Retrieval, Drexel University, Philadelphia, PA, USA, September 14-18, 2008, pages 263–268, 2008.

[17] Alexander Schätzle, Antony Neu, Georg Lausen, and Martin Przyjaciel-Zablocki. Large-scale bisimulation of RDF graphs. In Roberto De Virgilio, Fausto Giunchiglia, and Letizia Tanca, editors, *Proceedings of the Fifth Workshop on Semantic Web Information Management, SWIM@SIGMOD Conference 2013, New York, NY, USA, June 23, 2013*, pages 1:1–1:8. ACM, 2013.

[18] Michael Schmidt, Thomas Hornung, Georg Lausen, and Christoph Pinkel. Sp2bench: A SPARQL performance benchmark. *CoRR*, abs/0806.4627, 2008.

[19] Thanh Tran, Günter Ladwig, and Sebastian Rudolph. Managing structured and semistructured RDF data using structure indexes. *IEEE Trans. Knowl. Data Eng.*, 25(9):2076–2089, 2013.