Exploring RDF Graphs through Summarization and Analytic Query Discovery

Ioana Manolescu
ioana.manolescu@inria.fr
Inria, Institut Polytechnique de Paris
Palaiseau, France

ABSTRACT

Graph data is central to many applications, ranging from social networks to scientific databases. Graph formats maximize the flexibility offered to data designers, as they are mostly schema-less and thus can be used to capture very heterogeneous-structure content. RDF, the W3C's format for sharing open (linked) data, adds the possibility to attach *semantics* to data, describing application-domain constraints by means of ontologies; in turn, this leads to *implicit data* that is also part of a graph even if it is not explicitly in it.

In this paper, we present a structured walk through the problem of *analyzing and exploring RDF graphs by finding groups of structurally similar nodes*, and by *automatically identifying interesting aggregates theirein*. We outline the challenges raised by such processing in large, complex RDF graphs, outline the basic principles behind existing solutions, and highlight opportunities for future research.

1 INTRODUCTION

Graph data is increasingly popular, thanks to the flexibility it allows to its designers: it enables representing varying-structure entities together with their rich attributes and the relationships interconnecting them.

In particular, RDF graphs are abundantly present on today's Web, as RDF is the recommended format for sharing Open Data. The Linked Open Data Cloud Web site (https://lod-cloud.net/) lists numerous examples of RDF databases. Nevertheless, the multiplication of data sources is not sufficient to enable the construction of applications that take advantage of it. An important obstacle is rooted in the very advantages of RDF: its flexibility and the heterogeneity it tolerates in the data make it hard for users to *understand what a graph is about*, and potentially even harder to detect what is *interesting* within the graph.

Two approaches can be seen for analyzing and exploring a graph's content. On one hand, *node-focused exploration* could allow for instance users to identify a few nodes and/or edges they are interested in. This could be achieved by allowing them to search, e.g., through keywords, or by some statistical analysis, e.g., identifying nodes that are somehow outliers, through their content or through their structural properties. Such fine-granularity exploration enables gaining detailed knowledge about relatively small part of the graph. On the other hand, *group- or class-focused exploration* seeks to identify interesting subgraphs, or (most typically) groups of nodes, which are in a certain sense similar or comparable. The first step is thus to simplify the cognitive task of getting acquainted with a graph, by reducing it to the (simpler) task of understanding a smaller, abstract version thereof, where each group of nodes represents a "class" or "meta-node". Such a broad graph analysis may hide (or obscure within a larger group) interesting values or outliers, but it has the advantage of enabling a global, top-down view, which can be gained as one starts working with the graph.

The research highlighted below takes this second path. The problems to be solved are: how to efficiently build meaningful summaries of large RDF graphs (Section 2); and how to analyze and explore RDF graphs by means of aggregate queries (Section 3). Each problem raises specific conceptual and algorithmic challenges; we motivate the solutions we found, and point to interesting areas where the work could continue.

Are node groups an interesting metaphor for exploring RDF graphs? Figure 1 (from [8]) tends to suggest it. It depicts the properties of the subjects in a graph describing publications listed in the DBLP server. Each ring represents the frequency of a given RDF property among these subjects (or resources). Thus, the central blue ring reflects the property rdf:type, which clearly all the subjects have; the second one, dark blue, is date, which publications have, but authors do not; we can see a set of other properties present on almost all publications (their frequency diminishing as we move away from the center of the graph), while another set of resources have the name property but none of the properties that publications have; these are the authors.

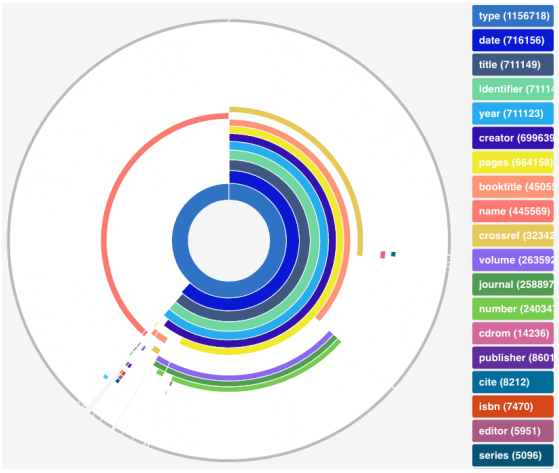

Figure 1: Property frequency analysis in an RDF graph [8].

2 SUMMARIZING RDF GRAPHS THROUGH STRUCTURAL QUOTIENTS

The problem of summarizing RDF graphs has been extensively studied, in particular drawing upon ideas and solutions proposed for summarizing generic graphs, or XML documents; RDF summarization approaches are surveyed in [3]. A brand of summaries well-established in database research is that of *structural quotients*: an equivalence relation is identified between the nodes

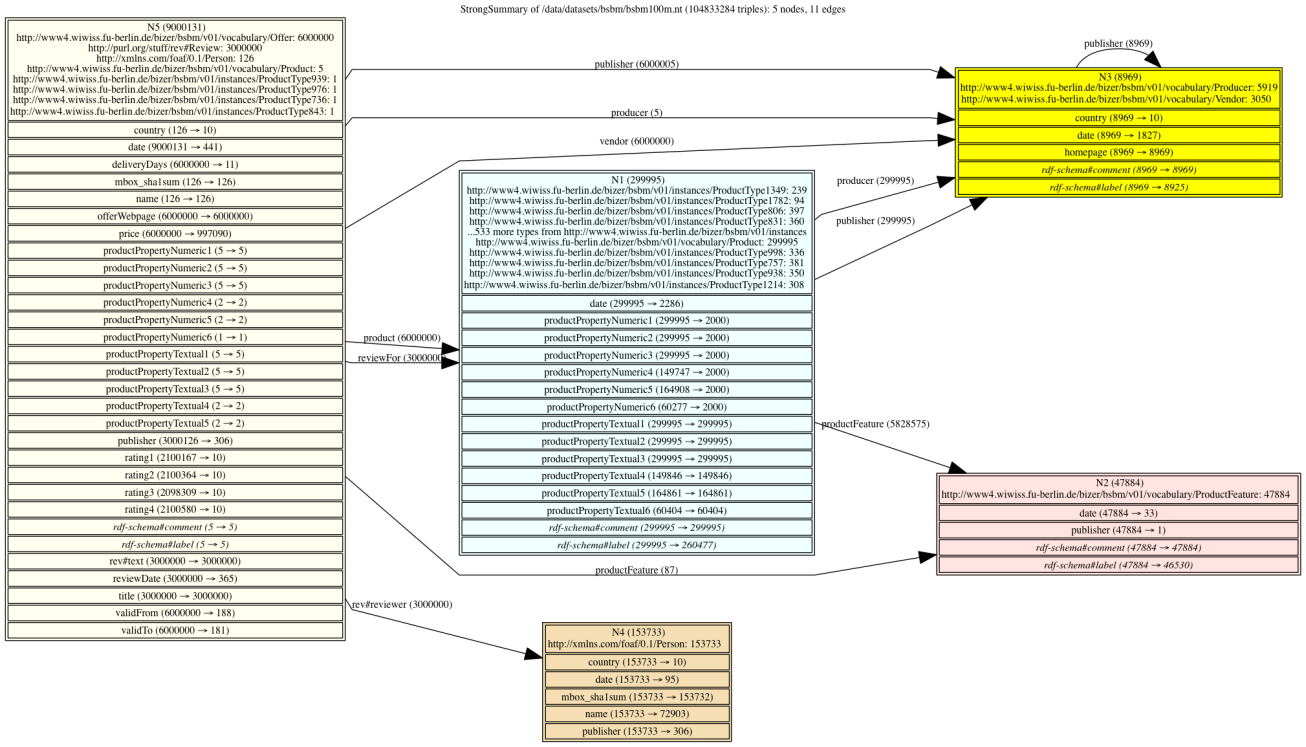

Figure 2: Sample Strong Summary [10] of a Berlin SPARQL Benchmark (BSBM) graph of 100 million triples. Other summary examples are depicted on the RDFQuotient project Web site, where our summarization tool is also available in open source.

of a graph, typically based on their incoming/outgoing edges. A quotient summary node has one node per equivalence class, and one edge between two summary nodes if and only if a corresponding edge connects, in the original graph, a pair of nodes they *represent*. Quotient summaries have been introduced as basis for structural indexing, in OEM databases [11] and subsequently for XML, e.g. [15].

The choice of an equivalence relation, thus, fully determines a summary. Which equivalence relation to pick? In [4, 10], we have proposed two novel relations, based on the transitive closure of sharing incoming, respectively, outgoing properties. Thus, if n_1, n_2 both have titles, n_2 and n_3 both have authors, while n_3, n_4 both have publication years, we say n_1 to n_4 share the same *outgoing property clique*, comprising the properties (edge labels) "title", "author", and "year". This outgoing clique (set of properties) is defined based on their (transitively) co-occurring on common nodes. Observe that n_1 and n_4 may be quite different from each other; in particular, they may have no property in common. Incoming property cliques are symetrically defined. Based on the notions of incoming, respectively, outgoing property cliques, we introduce two notions of equivalence, so-called *weak* and *strong* [10], and show that they lead to very compact summaries of RDF graphs for which previously proposed quotients lead to summaries having more nodes by orders of magnitude. Figure 2 illustrates this: the summary of a BSBM graph of 100 million triples has only 5 nodes and 11 edges, the size of a relatively simple Entity-Relationship diagram.

What sets the summarization of RDF graphs apart from related graph summarization problems? Several features concur.

First, RDF nodes may have *types*; this is encoded by graph edges, connecting the typed nodes to special kinds of resources in the graph, namely the type nodes themselves. A node may have zero, one, or more types, which may or may not be logically connected. Further, some nodes in a graph may have types, while others lack them. Types complicate summarization, since on one hand, they encapsulate precious application knowledge when present, but on the other hand, summarization must be able to make sense of a graph even in their absence. Therefore, we have distinguished *data-first* summarization, which groups nodes according to their types first and foremost, and then carries the type of a node to its representative in the summary. This is suited for graphs where types are mostly absent, or not sufficient to distinguish classes of nodes from each other. The opposite strategy is *type-first*; it groups nodes by their types, and only uses property cliques to differentiate between the untyped ones. Depending on the graph, *data-first* or *type-first* summarization may be more suitable in order to produce summaries easy to understand.

A second, more subtle aspect is due to the presence of an ontology, which may make part of the graph implicit, that is, triples may hold in the graph, which are not explicitly present there. In this case, summarizing the graph of explicit triples may not account for the implicit ones. We have proposed in [10] a sufficient condition under which one can compute the summary of a saturated graph (including all its implicit and explicit data), without actually saturating the graph; we also show that our Weak and Strong summaries, in their *data-first* incarnation, satisfy this condition, whereas any *type-first* summarization does not.

The summaries we devised, like many others, strive to separate nodes of a large graph in groups that simplify its understanding. Compared with other works, our goal has been to facilitate understanding at first sight the major groups of nodes in a graph. We made the hypothesis that accepting "transitively similar" nodes in a same group allows identifying such groups; our experiments bear out this claim. Another strong advantage of our summaries is that they can be all built in time linear in the size of the input [9, 10], including in *incremental* mode, that is, deriving the summary equivalence relation and summarizing the graph at the same time.

The compactness of our summaries comes at a cost of precision. For instance, they provide very poor support for indexing, since they are unable to guarantee that graph nodes represented by a certain summary node have, a certain property. More generally, they (and any other quotient summaries) reflect the structure, but not the values (leaf nodes) present in the graph.

3 EXPLORING RDF GRAPHS BY MEANS OF INTERESTING AGGREGATES

While aggregation is well-established as a way to analyze, aggregate and summarize relational data, the very meaning of aggregation has been slow-coming for graphs, and in particular for RDF. In March 2013, the SPARQL 1.1 specification introduced a Group-By primitive together with aggregation operators; their semantics is essentially lifted from the relational database world, and applied to the tuples of bindings resulting from the matches of a Where SPARQL block. Below we outline a path we started from devising an RDF counterpart to relational (data warehouse) relational queries, formalizing RDF analytical (aggregate) queries (Section 3.1), and (in subsequent, currently ongoing work) exploring RDF graphs by automatically identifying interesting aggregates (Section 3.2).

3.1 RDF aggregate queries

Our research [1, 6] considered, at about the same time, RDF aggregation at a *conceptual*: what should an *RDF analytical query* look like? The well-known concepts of facts, dimension and measure from the relational literature hardly fit. To start with, RDF graphs lack a previously-defined schema, and thus the *facts* at the heart of analytical processing are not defined; irregularity in the data may lead to a dimension or measure being absent, or being multiply defined. We proposed to define *RDF analytical (aggregate) queries* as a combination of a *fact query*, defining the set of resources to be treated like facts and analyzed together, a set of *dimension queries*, associating to each fact zero or more values against each dimension, a *measure query*, specifying what to use as a measurable property of each fact, finally an *aggregation function* among the usual ones (sum, max, average etc.) A sample aggregate query can be composed as follows:

- *Facts* are all the articles published between 2000 and 2020;
- A *dimension* is a country to which an authors' institutions are affiliated; many papers have authors from multiple countries, naturally leading to multiple values for a dimension;
- Another *dimension* is the year;
- A *measure* of a paper is a keyword in the paper abstract;
- The *aggregation function* counts the different keywords.

The analytical query described above groups papers by the year and author country, and for each paper group, it counts the keywords associated to papers published in that years with an author from that country.

As this example shows, a fact contributes to the answer of an RDF analytical query iff it has values for all dimensions and for the measure; a fact may contribute to several cells, if it has multiple values for one or several dimensions. This flexible model has numerous advantages for analyzing RDF graphs:

- There may be several *fact sets* in an RDF graph. One could, for instance, in a publication dataset, consider the articles to be the facts, and aggregate them according to their topics, their year of publication etc.; on the opposite - or rather, *at the same time* - one could consider the authors to be the facts, and articles (or the articles' years, or topics, or venues) as dimensions.
- As explained above, it flexibly accomodates the absence of a dimension or a measure, as well as their possible multiple values.

The formal semantics of such analytical queries [1, 6] is compatible with the SPARQL 1.1. aggregation semantics; the latter, however, is only concerned with the syntactic level, not with the more conceptual one where facts, dimensions and measures are specified.

3.2 Automatically identifying interesting RDF aggregates

As previously explained, RDF analytical queries enable expressing a large set of questions which enable characterizing, in a flexible manner, the nodes of an RDF graph. But what queries to ask?

A well-explored branch of research in relational data analytics concerns the automated identification of interesting analytical queries [17–19]. These works are placed in a typical relational data warehouse scenario, where a large number of dimensions exist, and seek to automatically proposed to the users the analytical queries that are likely to bring them most *insight*. For instance, in [18], a query is interesting (brings a useful insight) if it exhibits, on a subset of the facts, a trend that is different from the one that holds on the complete fact set.

In our Dagger project [8], we initiated an approach to automatically identify interesting analytical queries in RDF graphs. This was based on a set of simple choices:

- Chosing as facts all nodes of a given RDF type, or, alternatively, asking users to specify the fact query;
- Chosing as dimensions the properties that sufficiently many facts have, and whose number of distinct values does not exceed a certain threshold; we also introduced *derived* properties, such as the number of authors that a paper has, which we treat like a new propertu attached to the paper fact;
- Chosing a measure among the other (original or derived) properties of the facts;
- Considering an aggregate interesting if it maximizes a certain statistical measure of the aggregate query *result*.

Figure 3 illustrates the kinds of aggregates Dagger identified, in a set of DBLP publications from 1936 to 2006. At the top, the average number of authors of a published paper; we see the rise of co-authorship along the years. At the center, the number of published papers grouped by year; this graph really gives flesh to the concern that as an academic community we may be publishing too much! Last but not least, the graph at the bottom counts the books listed in DBLP and grouped by their publisher. The dominating bar corresponds to Springer; Infix Verlag comes second, and a set of bars at the left of the graph show different

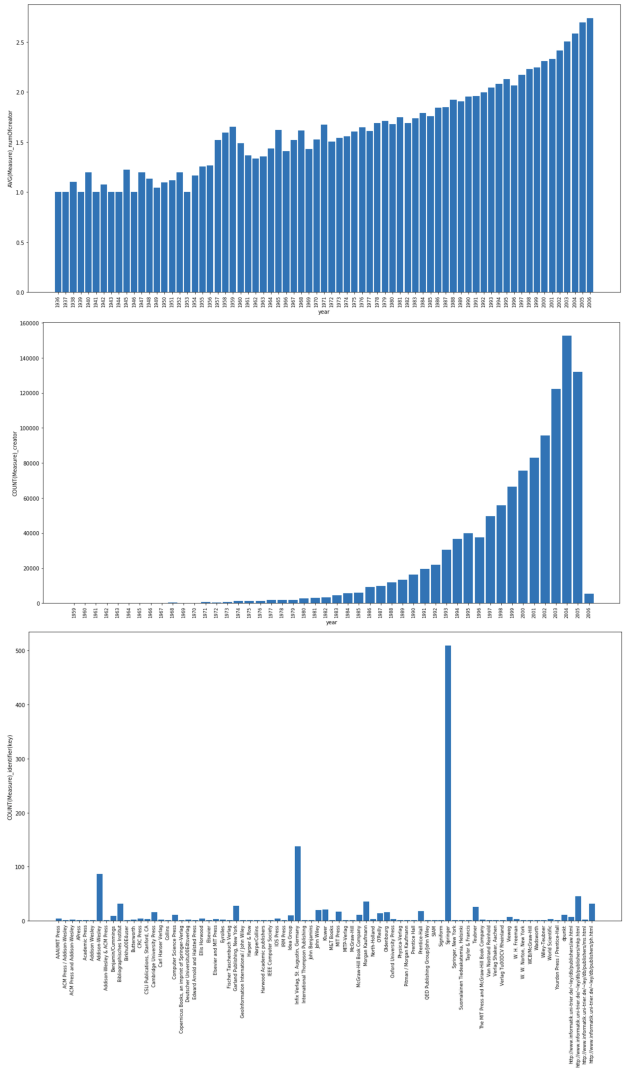

Figure 3: Interesting insights found by Dagger [8].

spellings for Addison-Wesley, leading to an artificial separation of their books over what would appear to be several publishers.

3.3 Scaling up the exploration of interesting aggregates

Dagger identifies interesting aggregates through exhaustive search: it explores and evaluates aggregates subject to a given time limit, before returning the most interesting ones. This made its exploration process lengthy. We explored in [14] the use of *sampling*, both to select the dimensions and measures to use for the facts, and to decide which aggregates are interesting. While this did reduce the running time, it provided no guarantee of the accuracy of the exploration thus abridged.

In the domain of relational data analytical processing, a key ingredient to the automated selection of interesting queries is the ability to explore many candidates, and discard as early as possible those queries which can be determined quickly enough to be not sufficiently interesting. *Online aggregation*, pioneered in [12] has been a crucial ingredient here: it allows to derive, while aggregate queries are being computed, an approximation (with a given confidence interval) of these queries' results. We have explored that path in Spade [7], our follow-up project on Dagger, where we make several new contributions:

- We enlarge the exploration space to *multidimensional* (not just mono-dimensional) aggregates;
- We introduce *more derived properties*, for instance by means of topic extractions from text; also, moving toward the generality of the analytical queries introduced in [6] we allow dimensions and measures to be defined by paths of a certain length starting in the facts;
- To cope with the expensive exploration of multidimensional aggregates while remaining efficient, we have devised a novel version of a well-known algorithm [20], capable of evaluating in a single pass all the aggregates determined by a set of dimensions, a measure and an aggregation function;
- Still toward the goal of scalability, we have devised novel *early-stop* techniques, capable of estimating the interestingness of an aggregation query while it is computed, and stop the computation as soon as it becomes clear that other aggregates, whose computation is ongoing, are more interesting.

Figure 4 illustrates a multidimensional aggregate Spade identified. It shows, for instance, that the "system" and "machine" keywords have been present from the early days of DBLP publications, whereas the "web" newcomer started its history in the 1990s; we see "system" making an important comeback (light yellow area toward the top right), idem for "network" etc.

Our work on Spade is ongoing at the time of this writing. We are still working to improve its performance, and to understand the interplay of the early-stop and of the multidimensional algorithms used to explore and estimate the interestingness of various RDF analytical queries.

4 CONCLUSION AND OUTLOOK

The field of graph analytics is by nature very broad, given the extreme diversity of data modeled as graphs. This paper summarizes a set of recent work carried with the global goal of helping users grasp the content of a large and potentially complex RDF graph. Our key findings can be summarized as follows:

- Identifying interesting node groups is an intuitive first step toward gaining an understanding of the graph, of its semantics, structure and content.
- The properties incoming and outgoing RDF resources can be used as a good basis for identifying such groups, provided that good measures are taken to avoid the extreme fragmentation which would result from requiring all nodes in a group to have *exactly* the same structure. Instead, summaries such as we introduced in [4, 9, 10] accept some heterogeneity among the nodes, which generally leads to easy-to-read summaries.
- If one also takes into account the values, that structural summaries completely disregard, there are many ways to explore how groups of nodes in RDF graphs compare among themselves, and countless combinations of facts, dimensions, and measures one could use. In Dagger [8] and its successor Spade [7], we are working to identify as quickly as possible interesting aggregates, with an interesting measure currently defined as the variance of the set of values that are part of the aggregate query result.

  Many avenues for future research are open.
- Personalization, user input, or query by example [13] could be blended with exploration such as we envisioned it, in order to help users get as soon as possible to the information they need for a specific task, in the spirit of [16].
- RDF graph semantics has not yet been fully taken into account in the exploration. It could be incorporated as a facet, or as a

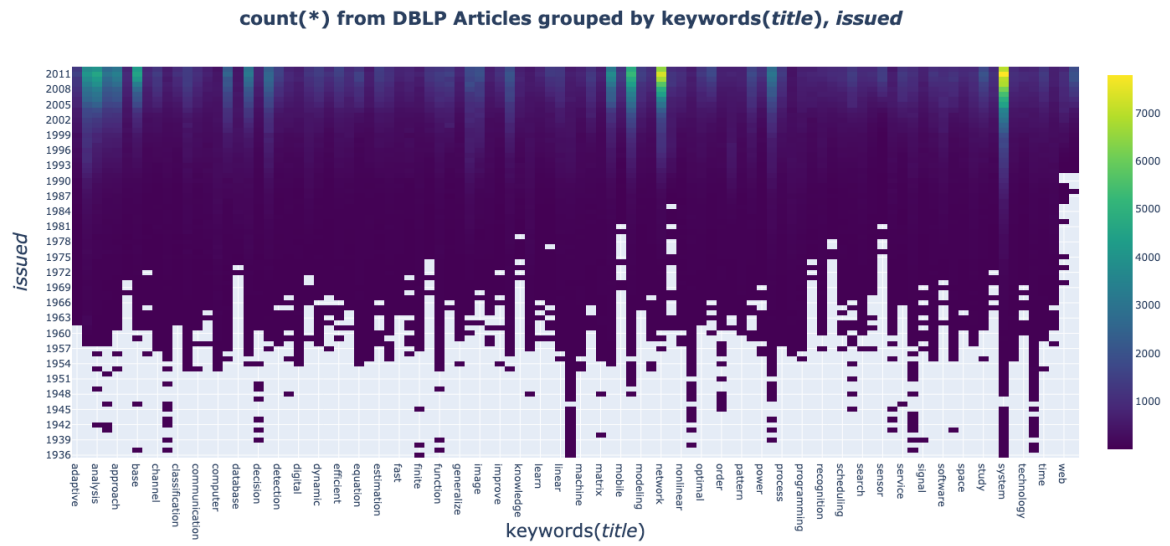

Figure 4: Sample interesting aggregate identified by SPADE: number of DBLP articles by years and keyword appearing in their titles. The darker the collor, the fewer articles there are.

way of navigating from one interesting insight to another one, on a closely (semantically) related set of items.

A related, if more mundane, question is which platform (or back-end) should best support such analytics; the competition among (RDF) graph processing platforms is currently hot, with no clear winner in sight. While many contenders exist, the very different kinds of processing envisioned, say, in Semantic Web integration queries, on one hand, and in social network analysis with the goal of influence maximization, on the other hand, make comparisons difficult, and convergence unlikely.

Going beyond exploration of RDF graphs, one could envision tools blending more strongly extraction of information from unstructured content, and structured data under one of its many forms. This kind of graphs are encountered, for instance, when integrating heterogeneous data sources such as those available to journalists. We have outlined such a graph-based integration framework in the CONNECTIONLENS [5] system. Such heterogeneous graphs exhibit even more structural and content heterogeneity; higher-levels abstraction methods are needed as a first step towards facilitating their understanding [2]. We plan to continue work on these topics, within the ANR SourcesSay project (2020-2024).

Acknowledgments This research has been partially funded by ANR-16-CE23-0010-01 and the H2020 research program under grant agreement nr. 800192. supported

REFERENCES

[1] Elham Akbari-Azirani, François Goasdoué, Ioana Manolescu, and Alexandra Roatis. 2015. Efficient OLAP Operations For RDF Analytics. In *International Workshop on Data Engineering meets the Semantic Web (DESWeb)*. Seoul, South Korea. https://doi.org/10.1109/ICDEW.2015.7129548

[2] Irène Burger, Ioana Manolescu, Emmanuel Pietriga, and Fabian Suchanek. 2020. Toward Visual Interactive Exploration of Heterogeneous Graphs. (2020).

[3] Sejla Cebiric, François Goasdoué, Haridimos Kondylakis, Dimitris Kotzinos, Ioana Manolescu, Georgia Troullinou, and Mussab Zneika. 2019. Summarizing Semantic Graphs: A Survey. *The VLDB Journal* 28, 3 (June 2019). https://hal.inria.fr/hal-01925496

[4] Šejla Čebirić, François Goasdoué, and Ioana Manolescu. 2015. Query-Oriented Summarization of RDF Graphs. In *Proceedings of the VLDB Endowment*, Vol. 8. Kohala Coast, Hawaii, United States. https://hal.inria.fr/hal-01178140

[5] Camille Chanial, Rédouane Dziri, Helena Galhardas, Julien Leblay, Minh-Huong Le Nguyen, and Ioana Manolescu. 2018. ConnectionLens: Finding Connections Across Heterogeneous Data Sources. *Proceedings of the VLDB Endowment (PVLDB)* 11 (2018), 4. https://doi.org/10.14778/3229863.3236252

[6] Dario Colazzo, François Goasdoué, Ioana Manolescu, and Alexandra Roatis. 2014. RDF Analytics: Lenses over Semantic Graphs. In *23rd International World Wide Web Conference*. Seoul, South Korea. https://doi.org/10.1145/2566486.2567982

[7] Yanlei Diao, Pawel Guzewicz, Ioana Manolescu, and Mirjana Mazuran. [n. d.]. Spade: A Modular Framework for Analytical Exploration of RDF Graphs. In *VLDB 2019 - 45th International Conference on Very Large Data Bases*. https://hal.inria.fr/hal-02152844

[8] Yanlei Diao, Ioana Manolescu, and Shu Shang. 2017. Dagger: Digging for Interesting Aggregates in RDF Graphs. In *International Semantic Web Conference (ISWC)*. Vienna, Austria. https://hal.inria.fr/hal-01577464

[9] François Goasdoué, Pawel Guzewicz, and Ioana Manolescu. 2019. Incremental structural summarization of RDF graphs. In *EDBT 2019 - 22nd International Conference on Extending Database Technology*. Lisbon, Portugal. https://hal.inria.fr/hal-01978784

[10] François Goasdoué, Paweł Guzewicz, and Ioana Manolescu. 2020. RDF graph summarization for first-sight structure discovery. *The VLDBJ Journal* (2020). To appear.

[11] Roy Goldman and Jennifer Widom. 1997. DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases. In *Proceedings of 23rd International Conference on Very Large Data Bases, 1997, Athens, Greece*. 436–445.

[12] Joseph M. Hellerstein, Peter J. Haas, and Helen J. Wang. 1997. Online Aggregation. In *SIGMOD*. 171–182.

[13] Matteo Lissandrini, Davide Mottin, Themis Palpanas, and Yannis Velegrakis. 2018. *Data Exploration Using Example-Based Methods*. Morgan & Claypool Publishers. https://doi.org/10.2200/S00881ED1V01Y201810DTM053

[14] Ioana Manolescu and Mirjana Mazuran. 2019. Speeding up RDF aggregate discovery through sampling. In *BigVis 2019 - 2nd International Workshop on Big Data Visual Exploration and Analytics*. Lisbon, Portugal. https://hal.inria.fr/hal-02065993

[15] Tova Milo and Dan Suciu. 1999. Index structures for path expressions. In *International Conference on Database Theory*. Springer, 277–295.

[16] Amit Somech, Tova Milo, and Chai Ozeri. 2019. Predicting "What is Interesting" by Mining Interactive-Data-Analysis Session Logs. In *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 2019*. 456–467. https://doi.org/10.5441/002/edbt.2019.42

[17] Bo Tang, Shi Han, Man Lung Yiu, Rui Ding, and Dongmei Zhang. 2017. Extracting Top-K Insights from Multi-dimensional Data. In *SIGMOD*. 1509–1524.

[18] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya G. Parameswaran, and Neoklis Polyzotis. 2015. SEEDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics. *PVLDB* 8, 13 (2015), 2182–2193.

[19] Yuhao Wen, Xiaodan Zhu, Sudeepa Roy, and Jun Yang. 2018. QAGView: Interactively Summarizing High-Valued Aggregate Query Answers. In *SIGMOD*. 1709–1712.

[20] Yihong Zhao, Prasad Deshpande, and Jeffrey F. Naughton. 1997. An Array-Based Algorithm for Simultaneous Multidimensional Aggregates. In *SIGMOD*. 159–170.