RDF-ANALYTICS: Interactive Analytics over RDF Knowledge Graphs

Maria-Evangelia Papadaki and Yannis Tzitzikas
Institute of Computer Science, FORTH-ICS, 70013 Heraklion, Greece, and
Computer Science Department, University of Crete, Greece
marpap@ics.forth.gr,tzitzik@ics.forth.gr

ABSTRACT

The formulation of structured queries in knowledge graphs is a challenging task that presupposes familiarity with the syntax of the query language and the contents of the knowledge graph. To alleviate this difficulty in this paper we introduce RDF-ANALYTICS, a novel system that enables plain users to formulate analytic queries over complex, i.e. not necessarily star-schema based, RDF knowledge graphs. To come up with an intuitive interface, we leverage the familiarity of users with Faceted Search (FS) systems, i.e. we extend FS with actions that enable users to formulate analytic queries, too. Distinctive characteristics of the approach is the ability to include arbitrarily long paths in the analytic query (accompanied with *count* information), interactive formulation of HAVING restrictions, the support of both Faceted Search (i.e. the locating of the desired resources in a faceted search manner) and analytic queries, and the ability to formulate nested analytic queries. Finally, we present the results of a preliminary task-based evaluation with users, which are very promising.

KEYWORDS

Knowledge Graphs, Analytics, Faceted Search

1 CONTEXT AND MOTIVATION

There are several Knowledge Graphs (KGs), i.e. collections of facts in the form "(subject, relation, object)" expressed in RDF, that integrate data from various sources: from general purpose, like DBpedia [3] and Wikidata [30], to domain specific repositories, e.g., Europeana [13], DrugBank [31], GRSF [28], ORKG [14], WarSampo [16], [5, 6] for Covid-19, and [7] for digital humanities. It would be very useful if plain users could analyze such interesting but complex amounts of data, interpret it, discover useful information and derive insights from it in an easy and flexible way.

2 CHALLENGES

Although, users can *browse* KGs through the provided web pages (in case dereferenceable URIs are supported), *search* them using keyword search (e.g. through [19]), or *query* them through plain SPARQL or through interactive query formulators (like [4] and [17]), there is not any standard method of formulating analytic queries. Indeed, the analysis of KGs in RDF is challenging since knowledge of the terminology and the syntax of query language are required. Such requirements are quite cumbersome for ordinary users and time-consuming for expert users.

2.1 Running Example

Suppose a KG with information about products and their related entities, e.g. companies, persons, locations, etc., with schema as shown in Figure 1 (for reasons of brevity namespaces are hidden).

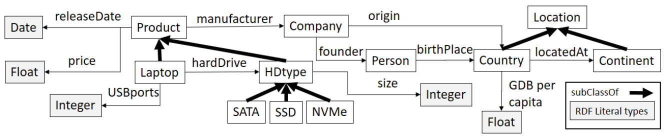

Figure 1: The schema of the running example

Assume that we would like to answer a query of the form like *"average price of laptops made in 2021 from US companies that have 2 USB ports and an SSD drive manufactured in Asia grouped by manufacturer"*. Such a query is quite complex, since it requires expressing several restrictions that also involve paths in the KG. Its expression in SPARQL is shown in Fig. 2.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
Prefix    ex:<http://www.ics.forth.gr/example#>
SELECT  ?m  (AVG(?p)  as ?avgprice)
WHERE {
?s rdf:type ex:Laptop.
?s ex:manufacturer ?m.
?m ex:origin ex:USA.
?s ex:price ?p.
?s  ex:USBPorts ?u.
?s ex:hardDrive ?hd.
?hd rdf:type ex:SSD.
?hd ex:manufacturer ?hdm.
?hdm ex:origin ?hdmc.
?hdmc ex:locatedAt ex:Asia.
FILTER (?u >= 2).
?s ex:releaseDate ?rd .
FILTER ( ?rd >= "2021-01-01T00:00:00"^^xsd:dateTime  &&
?rd <= "2021-12-31T00:00:00"^^xsd:dateTime)
} GROUP BY ?m
```

Figure 2: Expression in SPARQL of the query of §2.1.

3 DIRECTION AND APPROACH

To alleviate the aforementioned difficulty, we propose an interaction model that allows plain users to compose analytic queries through simple clicks (or simple selections), while exploring the contents of a KG even if they have no technical background. We aim at finding a generic interaction model that can be applied to any RDF dataset (not only to datasets that have a star schema) and that guides users to create only answerable queries, "protecting" them from spending effort on trying to formulate queries that are not answerable due to lack of data.

In particular, we leverage the familiarity of users with *Faceted Search (FS)* [23], since this model lets users express complex conditions through simple clicks. We start from a general model for faceted search over RDF data [27] and we extend it with actions that enable formulating analytic queries. The proposed model supports only answerable queries, restrictions, HAVING clauses, nested queries, paths in both FS and analytic queries, while it provides count information in any state.

Specifically, the classical FS interface usually comprises two main frames: the left one that is used for the facets (filters), and the right one that is used for showing the objects, see Figure 3(left). We propose enriching the user actions of the left frame with actions for formulating analytic queries, specifically notice the G and ± buttons on each facet shown in Figure 3(right): the first for specifying grouping function(s), the second for the measuring function(s). In addition, we need one additional frame, let call it Answer Frame, for short *AF*, for showing the results of the analytic query (in a tabular or other method). To enable the formulation of HAVING restrictions we propose a button "Explore with FS" in the Answer Frame, through which the user can load the results of the current query as a new dataset, and can (again through FS) specify restrictions. The latter restrictions correspond to HAVING restrictions over the original dataset.

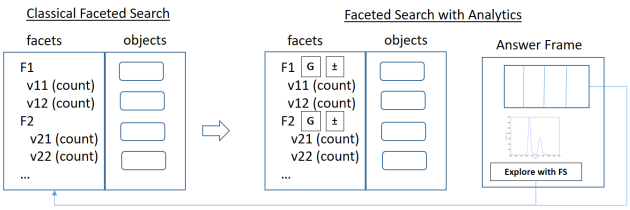

Figure 3: The core elements of the GUI for Faceted Search and Analytics

4 THE SYSTEM RDF-ANALYTICS

We have implemented the aforementioned ideas as a web application, that we call RDF-ANALYTICS. The server-side uses the triplestore Virtuoso[1] that offers persistent storage and SPARQL endpoint, while the front-end side of the system was based on Angular[2].

For example, for formulating the query in the running example (§2.1), i.e. *"average price of laptops made in 2021 from US companies that have 2 USB ports and an SSD drive manufactured in Asia grouped by manufacturer"*, the user has to express in a FS manner the condition "laptops made in 2021 from US companies that have 2 USB ports and an SSD drive manufactured in Asia", and use one button for specifying the "grouped by" and another for specifying the "avg price".

Figure 4 shows a screenshot of RDF-ANALYTICS: On the left of each facet name, there is an expansion icon, i.e. ">", enabling the user to see the top-level sub-classes and the applicable properties. On the right of each facet name, there is a check-box and three buttons:

[1]http://docs.openlinksw.com/virtuoso/
[2]https://angular.io/

- ☐: for filtering the results (through values of that facet)
- G: for grouping the results (with respect to that facet)
- ±: for selecting the function, i.e. avg, min, max, etc, that will be applied to each group of the analytic query
- > : for expanding a property path (unlimited depth, bidirectional)

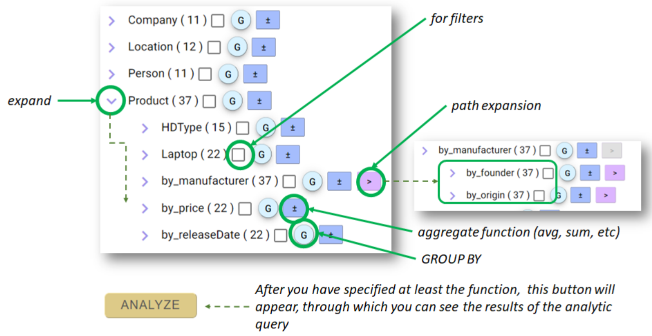

Figure 4: Basic buttons of RDF-ANALYTICS

Now Figure 5 shows how the user can restrict the *numeric values* of a facet within intervals by specifying the minimum and maximum values of them, and how *derived attributes* e.g. YEAR, MONTH, DAY of a Date, can be extracted. For example, in the running example where the user wants to group the laptops by year, (s)he would click on the grouping button that is next to the facet of the "date" and then (s)he would select the derived attribute of "year" from the provided menu.

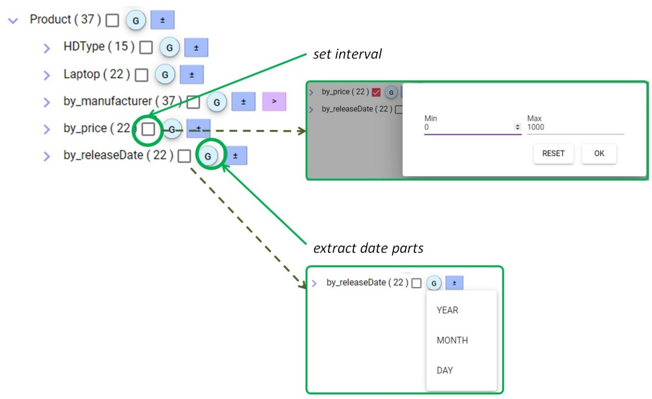

Figure 5: Setting intervals

The results of an analytic query are presented in a tabular form and as a plot as shown in Figure 6 for the query of the running example. It is important to stress that if the user clicks on the button "Explore with FS" that is provided below the analytical results, (s)he can also load them, as a new dataset (as shown in the bottom part of Figure 6). Then (s)he can proceed in formulating restrictions and group by queries. This enables the formulation of HAVING restrictions (and nested queries of unlimited depth).

5 EXPRESSIVENESS

Our main objective is to cover common needs in a familiar interaction style, not to propose an interaction model with very high expressive power but too complex to use. Nevertheless, the

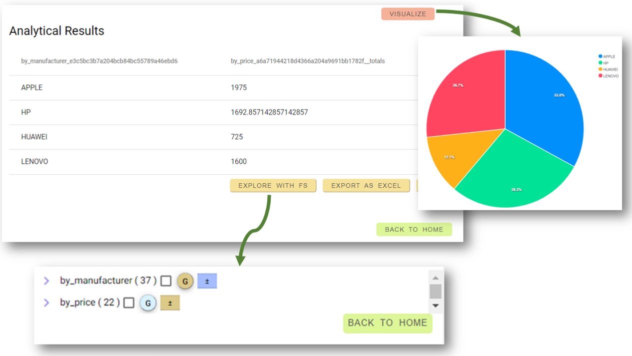

Figure 6: Presentation and visualization of the results and loading them as a new dataset

resulting model is very expressive enabling the expression of analytic queries that involve complex restrictions that involve property paths. Furthermore, the ability to load the results of a query, as a new dataset over which the user can continue query formulation, enables the formulation of analytic queries with HAVING clause in an intuitive manner.

From the perspective of *OLAP operations* [29], i.e. *roll up* (aggregate data by ascending concept hierarchy), *drill-down* (navigate from less detailed data to more detailed data), *slice* (perform a selection on one dimension of the given cube), *dice* (describe a sub-cube by operating a selection on two or more dimensions), *pivot* (provide an alternative presentation of the data), the interaction of RDF-ANALYTICS supports all of them. In particular, traversing up the hierarchy of a facet corresponds to roll-up, traversing down the hierarchy of a facet corresponds to drill-down, picking one value for a facet corresponds to slice, picking two or more values from multiple facets corresponds to dice, and moving to a facet which is directly or indirectly connected to the facet of focus corresponds to pivot.

6 EFFICIENCY

The information required by the interaction model and the analytics is gained through SPARQL, i.e. the system gradually builds the SPARQL query that will be sent to the SPARQL endpoint. This enables the application of the system to various endpoints. The various techniques that have been proposed in the literature for the optimization of SPARQL analytic queries, e.g. [11, 12] could be investigated for applying them to our system as well; this is a topic that goes beyond the scope of the current demo paper.

7 RELATED WORK AND NOVELTY

In general, there are not so many works, neither running systems (for a recent survey, see [21]).

There are a few works that support the formulation of analytic queries directly over RDF, for instance [9] supports guided query building (including analytic queries) with an implementation over the SPARKLIS editor [8]. As regards expressiveness, HAVING-restrictions are not supported, neither count information during query formulation, reducing in this way that exploratory characteristics of the process. In addition, the GUI is not the classical of FS, so it is not familiar to everyone. Another work that falls in this category is [25] that describes a possible extension of SemFacet [15] to support numeric value ranges and aggregation. That paper investigates theoretical query management aspects, it lacks an interface and implementation. Moreover, that model does not support explicit path expansion; instead the authors use the notion of "recursion" to capture reachability-based facet restrictions. Towards the direction of our work (analytics directly over RDF), [20] analyzed the applicability of an abstract language for analytics (HIFUN [26]) over RDF, and provided the algorithms for translating HIFUN queries to SPARQL queries. However, the interactive formulation of a HIFUN query is missing from that work. In the current work we want to fill this gap, since in knowledge graphs with broad coverage it is difficult to find and select the right property let alone the formulation of restrictions.

Another direction is the *definition of a data cube over RDF*, i.e. there are works that implicitly define a data cube over existing RDF graphs[3] [1, 2, 32], and then apply OLAP. One weakness of this approach (as stressed also in [9]) is that is requires someone with technical knowledge to define the required data cube(s). Apart from reduced flexibility, the user cannot leverage the wealth of connections of the knowledge graph, since the user is restricted on the data cube.

Another related topic is the *publishing of statistical data*, specifically the adoption of the RDF data cube vocabulary[4] for publishing and exchanging statistical data using the W3C RDF standard (e.g. [24]).

There are also, *domain specific works* (focusing on a particular topic, not on any RDF dataset), like [10] that motivates knowledge graph-enabled cancer data analytics, or [18] that describes an analogous work for covid-19 related data. Such works describe domain-specific pipelines for constructing the desired knowledge graph, for supporting particular analytic queries and visualizations. Such works do not aim at providing general-purpose methods for knowledge graph analytics.

7.1 Our position and contribution

We presented an approach with the following key characteristics: (i) it guides the user in query formulation, and the process never leads to empty results, (ii) it supports both FS and analytic queries, (iii) it supports HAVING clauses, (iv) it supports counts and paths in both FS and analytic queries, and (v) it leverages the familiarity of users with FS.

8 EVALUATION

Comparison with related systems. In Table 1, we compare the two most related systems to our approach, according to some important functionalities, i.e. applicability (application on star schemas or over any RDF graph), support of basic analytic queries, support of analytic queries with HAVING, support of plain Faceted Search, support of property paths in faceted search and analytics, visualization, as well as if there are running systems, and if they have been evaluated. We can notice that RDF-ANALYTICS has the highest number of supported features.

Task-based Evaluation with Users. We performed a small-scale evaluation with users to investigate if they can formulate easily analytic queries (especially queries that include path expressions), and to gain a general feedback from them. We defined 10 tasks and 10 users have participated, so far. We did not train them; we just provided them a concise help page explaining the actions of the buttons. The results so far are very promising in

[3] https://team.inria.fr/oak/projects/warg/
[4] https://www.w3.org/TR/vocab-data-cube/

Table 1: Comparing the functionalities of related systems

System	Applicability (STAR vs ANY)	Analytic queries: basic	Analytic queries: with Having	Plain Faceted Search	Property Paths (in Faceted Search and analytics)	Visualization	Running system	Evaluation
[25]	ANY	Yes	Yes	Yes but with No Count information	Not explicitly, reachability	No	No	No
[9]	ANY	Yes	No	No. Special interface	Not clear	No	Yes	Yes
Our approach	ANY	Yes	Yes by AF	Yes	Yes with counts	No	Yes	Yes

terms of task completion (success 73%, partial success 2%, fail 25%) and user rating (Very useful 50%, Useful 50%, Little Useful 0%, Not Useful 0%). We plan to perform a more extending evaluation (with more users and more tasks) of the system when it will be enriched with with extra visualization capabilities.

9 CONCLUDING REMARKS

By implementing `RDF-ANALYTICS` we demonstrated the feasibility of exploring and formulating analytic queries over arbitrary knowledge graphs, without presupposing knowledge neither of the terminology of the data, nor of the query language. Distinctive characteristics of the presented approach is that (i) it *guides* the user in query formulation, and the formulation process never leads to empty results, (ii) it supports *both* Faceted Search and analytic queries, (iii) it supports complex restrictions and path expressions, (iv) it supports HAVING restrictions and nested queries, (v) it leverages the familiarity of users with FS, and (vi) it can be applied directly over a SPARQL endpoints. The results of the small-scale evaluation with users provided evidence that the users can use this approach to formulate analytic queries.

The deployment of the system used is accessible through http://62.217.127.128:8080/Interactive-RDF-Analytics/ and is under continuous improvement. In future we plan to investigate the extension of the model with *transformation operators* that the user would apply at interaction time, for handling empty values and multi-valued properties. In addition, we plan to investigate issues related to optimizations for efficiency.

Acknowledgments. This work has received funding from the European Union's Horizon 2020 coordination and support action 4CH (Grant agreement No 101004468).

REFERENCES

[1] Elham Akbari Azirani, François Goasdoué, Ioana Manolescu, and Alexandra Roatiş. 2015. Efficient OLAP operations for RDF analytics. In *2015 31st IEEE International Conference on Data Engineering Workshops*. IEEE, 71–76.
[2] Boualem Benatallah, Hamid Reza Motahari-Nezhad, et al. 2016. Scalable graph-based OLAP analytics over process execution data. *Distributed and Parallel Databases* 34 (2016).
[3] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia-A crystallization point for the Web of Data. *Journal of web semantics* 7, 3 (2009), 154–165.
[4] Gonzalo Diaz, Marcelo Arenas, and Michael Benedikt. 2016. SPARQLByE: Querying RDF data by example. *Proceedings of the VLDB Endowment* 9, 13 (2016), 1533–1536.
[5] Dimitar Dimitrov, Erdal Baran, Pavlos Fafalios, Ran Yu, Xiaofei Zhu, Matthäus Zloch, and Stefan Dietze. 2020. TweetsCOV19–A Knowledge Base of Semantically Annotated Tweets about the COVID-19 Pandemic. In *29th ACM International Conference on Information and Knowledge Management (CIKM 2020)*.
[6] Lucy Lu Wang et al. 2020. Cord-19: The covid-19 open research dataset. https://doi.org/10.48550/ARXIV.2004.10706
[7] Pavlos Fafalios, Kostas Petrakis, Georgios Samaritakis, Korina Doerr, Athina Kritsotaki, Yannis Tzitzikas, and Martin Doerr. 2021. FAST CAT: Collaborative Data Entry and Curation for Semantic Interoperability in Digital Humanities. *ACM Journal on Computing and Cultural Heritage* 14 (2021). Issue 4.
[8] Sébastien Ferré. 2017. Sparklis: An expressive query builder for SPARQL endpoints with guidance in natural language. *Semantic Web* 8, 3 (2017), 405–418.
[9] Sébastien Ferré. 2021. Analytical Queries on Vanilla RDF Graphs with a Guided Query Builder Approach. In *International Conference on Flexible Query Answering Systems*. Springer, 41–53.
[10] SM Shamimul Hasan, Donna Rivera, Xiao-Cheng Wu, Eric B Durbin, J Blair Christian, and Georgia Tourassi. 2020. Knowledge graph-enabled cancer data analytics. *IEEE journal of biomedical and health informatics* 24, 7 (2020), 1952–1967.
[11] Dilshod Ibragimov, Katja Hose, Torben Bach Pedersen, and Esteban Zimányi. 2015. Processing aggregate queries in a federation of SPARQL endpoints. In *European Semantic Web Conference*. Springer, 269–285.
[12] Dilshod Ibragimov, Katja Hose, Torben Bach Pedersen, and Esteban Zimányi. 2016. Optimizing aggregate SPARQL queries using materialized RDF views. In *International Semantic Web Conference*. Springer, 341–359.
[13] Antoine Isaac and Bernhard Haslhofer. 2013. Europeana linked open data–data. europeana. eu. *Semantic Web* 4 (2013).
[14] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture*. 243–246.
[15] Evgeny Kharlamov, Luca Giacomelli, Evgeny Sherkhonov, Bernardo Cuenca Grau, Egor V Kostylev, and Ian Horrocks. 2017. Semfacet: Making hard faceted search easier. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2475–2478.
[16] Mikko Koho, Esko Ikkala, Petri Leskinen, Minna Tamper, Jouni Tuominen, and Eero Hyvönen. 2020. WarSampo Knowledge Graph: Finland in the Second World War as Linked Open Data. *Semantic Web – Interoperability, Usability, Applicability* (2020). https://doi.org/10.3233/SW-200392 In press.
[17] Vangelis Kritsotakis, Yannis Roussakis, Theodore Patkos, and Maria Theodoridou. 2018. Assistive Query Building for Semantic Data.. In *SEMANTICS Posters&Demos*.
[18] Franck Michel, Fabien Gandon, Valentin Ah-Kane, Anna Bobasheva, Elena Cabrio, Olivier Corby, Raphaël Gazzotti, Alain Giboin, Santiago Marro, Tobias Mayer, et al. 2020. Covid-on-the-Web: Knowledge graph and services to advance COVID-19 research. In *International Semantic Web Conference*. Springer, 294–310.
[19] Christos Nikas, Giorgos Kadilierakis, Pavlos Fafalios, and Yannis Tzitzikas. 2020. Keyword Search over RDF: Is a Single Perspective Enough? *Big Data and Cognitive Computing* 4, 3 (2020), 22.
[20] Maria-Evangelia Papadaki, Nicolas Spyratos, and Yannis Tzitzikas. 2021. Towards Interactive Analytics over RDF Graphs. *Algorithms* 14, 2 (2021), 34.
[21] Maria-Evangelia Papadaki, Yannis Tzitzikas, and Michalis Mountantonakis. 2023. A Brief Survey of Methods for Analytics over RDF Knowledge Graphs. *Analytics* 2, 1 (2023), 55–74.
[22] F. Gandon R. Gazzotti, F. Michel. 2020. CORD-19 Named Entities Knowledge Graph (CORD19-NEKG).
[23] Giovanni Maria Sacco and Yannis Tzitzikas. 2009. *Dynamic taxonomies and faceted search: theory, practice, and experience*. Springer.
[24] Percy E Rivera Salast, Michael Martin, Fernando Maia Da Mota, Sören Auer, Karin K Breitman, and Marco A Casanova. 2012. Olap2datacube: An ontowiki plug-in for statistical data publishing. In *2012 Second International Workshop on Developing Tools as Plug-Ins (TOPI)*. IEEE, 79–83.
[25] Evgeny Sherkhonov, Bernardo Cuenca Grau, Evgeny Kharlamov, and Egor V Kostylev. 2017. Semantic faceted search with aggregation and recursion. In *International Semantic Web Conference*. Springer, 594–610.
[26] Nicolas Spyratos and Tsuyoshi Sugibuchi. 2018. HIFUN-a high level functional query language for big data analytics. *Journal of Intelligent Information Systems* 51 (2018).
[27] Yannis Tzitzikas, Nikos Manolis, and Panagiotis Papadakos. 2017. Faceted exploration of RDF/S datasets: a survey. *Journal of Intelligent Information Systems* 48, 2 (2017), 329–364.
[28] Yannis Tzitzikas, Yannis Marketakis, Nikos Minadakis, Michalis Mountantonakis, Leonardo Candela, Francesco Mangiacrapa, et al. 2019. Methods and Tools for Supporting the Integration of Stocks and Fisheries. In *Chapter in Information and Communication Technologies in Modern Agricultural Development, Springer, 2019*. Springer.
[29] Panos Vassiliadis and Timos Sellis. 1999. A survey of logical models for OLAP databases. *ACM Sigmod Record* 28, 4 (1999), 64–69.
[30] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.
[31] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. 2018. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic acids research* 46, D1 (2018), D1074–D1082.
[32] Peixiang Zhao, Xiaolei Li, Dong Xin, and Jiawei Han. 2011. Graph cube: on warehousing and OLAP multidimensional networks. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*.