An RDF Dataset Generator for the Social Network Benchmark with Real-World Coherence

Mirko Spasić[1,2], Milos Jovanovik[1,3], and Arnau Prat-Pérez[4]

[1] OpenLink Software, UK
[2] Faculty of Mathematics, University of Belgrade, Serbia
[3] Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University in Skopje, Macedonia
[4] Sparsity Technologies, Spain
{mspasic, mjovanovik}@openlinksw.com, arnau@sparsity-technologies.com

Abstract. Synthetic datasets used in benchmarking need to mimic all characteristics of real-world datasets, in order to provide realistic benchmarking results. Synthetic RDF datasets usually show a significant discrepancy in the level of structuredness compared to real-world RDF datasets. This structural difference is important as it directly affects storage, indexing and querying. In this paper, we show that the synthetic RDF dataset used in the Social Network Benchmark is characterized with high-structuredness and therefore introduce modifications to the data generator so that it produces an RDF dataset with a real-world structuredness.

Keywords: Data Generation, Social Network Benchmark, Synthetic Data, Linked Data, Big Data, RDF, Benchmarks.

1 Introduction

Linked Data and RDF benchmarking require the use of real-world or synthetic RDF datasets [2]. For better benchmarking, synthetic RDF datasets need to comply with the general characteristics observed in their real-world counterparts, such as the schema, structure, size, distributions, etc. As the authors of [3] show, synthetic RDF datasets used in benchmarking exhibit a significant structural difference from real datasets: real-world RDF datasets are less coherent, i.e. have a lower degree of structuredness than synthetic RDF datasets. This structural difference is important as it has direct consequences on the storage of the data, as well as on the ways the data are indexed and queried.

In order to create the basis for more realistic RDF benchmarking, we modify the existing RDF data generator for the Social Network Benchmark so that the resulting synthetic dataset follows the structuredness observed in real-world RDF datasets. Additionally, we implement the coherence measurement for RDF datasets as a Virtuoso procedure, to simplify the process of measuring the structuredness of any given RDF dataset by using the RDF graph stored in the quad store, instead of using RDF files.

* The presented work was funded by the H2020 project HOBBIT (#688227).

2 Background and Related Work

2.1 Data Generator for the Social Network Benchmark

The Social Network Benchmark (SNB) [4] provides a synthetic data generator (Datagen)[1], which models an online social network (OSN), like Facebook. The data contains different types of entities and relations, such as persons with friendship relations among them, posts, comments or likes. Additionally, it reproduces many of the structural characteristics observed in real OSNs, summarized below.

Attribute correlations. Real data is correlated by nature. For instance, given names to persons are correlated with their country. Similarly, the textual content of a message is correlated with the interests of its creator. For the purpose of benchmarking and performance evaluation, reproducing these correlations is essential since their understanding can be used to generate optimal execution plans or to properly lay out the data in memory. Datagen uses dictionaries extracted from DBpedia to create entities with correlated attribute values.

Degree distributions. The underlying graph structure of OSNs exhibits skewed degree distributions: most of people have between 20 to 200 friends, while a few of them have over 5000 friends [12]. In a real database system, this skewness complicates the estimation of cardinalities required to produce optimal execution plans, or the load balancing when executing parallel operators over adjacency lists. Also, nodes with a large degree, commonly known as hubs, must be carefully handled, specially in distributed systems, where traversing these nodes can incur in a large inter-node communication. Datagen takes the degree distribution of Facebook and empirically reproduces it.

Structure-Attribute correlations. The homophily principle states that in a real social network, similar people have a larger probability to be connected, which leads to the formation of many transitive relations between connected people with similar characteristics. As a consequence, even though it is commonly accepted that graph data access patterns are usually random, in practice there is some degree of locality that can be exploited. For instance, people from a given country are more likely to be connected among them than to people from other countries. Thus, this information can be used to lay out data in memory wisely to improve graph traversal's performance, for instance, by putting all people in a country closer in memory. Datagen generates person relationships based on different correlation dimensions (i.e. the interests of a person, the place that person studied, etc.). These correlation dimensions are used to sort persons in such a way that those that are more similar, are placed close. Then, edges are created between close by persons, with a probability that decreases geometrically with their distance. As shown in [11], this approach successfully produces attribute correlated networks with other desirable characteristics such as large clustering coefficient, a small diameter or a large largest connected component.

Spiky activity volume. In a real social network, the amount of activity is not uniform but reactive to real-world events. If a natural disaster occurs, we

[1] `https://github.com/ldbc/ldbc_snb_datagen`

will observe people talking about it mostly after the time of the disaster, and the associated activity volume will decay as the hours pass. This translates to an spiky volume activity along the like of the social network, mixing moments with a high load with situations where the load level is small. Also, this means that the amount of messages produced by people and their topics are correlated with given points in time, which complicates the work of a query optimizer when estimating cardinalities for their query plans. This can also make a system unable to cope with the load if it has not been properly overprovisioned to handle these spiky situations. Instead of generating posts and comments uniformly distributed along time, Datagen creates virtual events of different degrees of relevance, which are used to drive the generation of the user activity, producing a spiky distribution.

2.2 Dataset Coherence

The comparison of data generated with existing RDF benchmarks (TPC-H, BSBM, LUBM, SP2Bench, etc.) and data found in widely used real RDF datasets (DBpedia, UniProt, etc.) shows that these two have significant structural differences [3]. In the same paper, the authors introduce a composite metric, called *coherence* of a dataset, in order to quantify the structuredness of the dataset D with respect to the type system $\mathcal{T}$ as follows:

$$CH(\mathcal{T}, D) = \sum_{T \in \mathcal{T}} WT(CV(T, D)) \cdot CV(T, D)$$

This is the weighted sum of the coverage $CV(T, D)$ of individual types $T \in \mathcal{T}$, where the weight coefficient $WT(CV(T, D))$ depends on the number of properties for a type T, the number of entities in dataset D of type T, and their share in the totality of the dataset D among the other types. Its rationale is to give higher impact to types with more instances and properties. $CV(T, D)$ represents the coverage of type T on the dataset D. It depends on whether the instances of the type T set a value for all its properties. If that is the case for all the instances, the coverage will be 1 (perfect structuredness), otherwise it will take a value from $[0, 1)$. The conclusion of [3] is that there is a clear distinction in the structuredness, i.e. the coherence between the datasets derived from the existing RDF benchmarks and the real-world RDF datasets. For the first ones, the values range between 0.79 (for SP2Bench) and 1 (for TPC-H) showing us the characteristics of relational databases, while the coherence values for almost all real-world datasets are below or around 0.6.

3 Measuring RDF Dataset Coherence in Virtuoso

In order to make an RDF benchmark more realistic, the dataset should follow the nature of real data. Since the SNB dataset is developed to test not only RDF stores, but also graph database systems, graph programming frameworks, relational and noSQL database systems, we wanted to measure how this dataset is suitable for RDF benchmarks.

The authors of [3] propose a workflow to compute the coherence of a dataset in a few steps: assembling all triples into a single file, data cleaning and normalization, generating several new files and sorting them in different orders to provide the ability that the corresponding metrics can be collected by making a single pass of the sorted file. The disadvantages of this approach are the memory requirements for storing all files in non-compressed format, and the time required to sort them. Also, the sorting process could use additional temporary space which is not negligible.

Here, we propose a new approach to compute coherence of any dataset, using an efficient RDF store, such as Virtuoso [5], leaving to the system to take care of the efficacy and data compression. Virtuoso is a column store with good compression capabilities, thus we will have a simpler and much more space- and time-efficient procedure for calculating the proposed metric. First, we load the dataset in question in a graph within Virtuoso, with a single command (*ld_dir*). Afterwards, we define a stored procedure in Virtuoso/PL for calculating the coherence, by selecting all types from a dataset, calculating their individual coverage and the weighted sum coefficient. The procedure, along with its supporting procedures, is available on GitHub[2]. Here, we show the *coverage()* procedure:

```
create procedure coverage (in graph VARCHAR, in t LONG VARCHAR) {
  declare a, b, c bigint;
  select sum(cnt) into a from (
       select t2.P as pred, count(distinct t1.S) as cnt
       from RDF_QUAD t1, RDF_QUAD t2
       where t1.S = t2.S and t2.P <> iri_to_id('rdf:type')
         and t1.G = iri_to_id(graph) and t2.G = iri_to_id(graph)
         and t1.P = iri_to_id('rdf:type') and t1.O = iri_to_id(t)
       group by pred
  ) tmp ;
  select count(distinct t2.P) into b from RDF_QUAD t1, RDF_QUAD t2
  where t1.S = t2.S and t2.P <> iri_to_id('rdf:type')
    and t1.G = iri_to_id(graph) and t2.G = iri_to_id(graph)
    and t1.P = iri_to_id('rdf:type') and t1.O = iri_to_id(t) ;
  select count(distinct S) into c from RDF_QUAD
  where G = iri_to_id(graph)
    and P = iri_to_id('rdf:type') and O = iri_to_id(t) ;
  return cast (a as real) / (b * c);
}
```

Using the original SNB Datagen, we prepared the datasets whose sizes and coherence metrics are presented in the Section 5. Since their coherence varies from 0.86 to 0.89, we can conclude that these datasets are much more structured than the real-world RDF datasets, thus they are not suitable for benchmarking RDF stores. Our intention is to make them mimic real-world Linked Data datasets, with a structuredness level of around 0.6.

[2] https://github.com/ldbc/ldbc_snb_implementations

4 A Realistic RDF Dataset Generator for the Social Network Benchmark

The original SNB Datagen (Section 2.1) reproduces the important structural characteristics observed in real online social networks. However, the authors of [3] show that structuredness is also important for RDF datasets used for benchmarking. Therefore, we modify the SNB Datagen so that we lower the structuredness, i.e. coherence measure, from around 0.88 to around 0.6, to comply with the structuredness of real-world RDF datasets.

The authors of [3] propose a generic way of decreasing the coherence metric for any dataset, without using domain specific knowledge. The consequence of this modification is the reduction of the dataset size. We decided to take a different approach and make some changes to the initial data generator, introducing new predicates for some instances, as well as removing some triples from the initial dataset, all while taking into account the reality of the specific domain, i.e. a social network. This enrichment phase provides a more realistic and complex dataset, complying with the current state and features of real-world social networks. In Table 1, we present the most dominant types from the SNB dataset, along with their weights. We omit the other types, as their weights are not significant in this case.

Table 1. Entity types and their weights from the SNB dataset.

Type	Weight
Comment	0.6477
Post	0.3126
Forum	0.0282
Person	0.0031

The weight of a type mostly depends on the number of its instances in the dataset. In the SNB dataset the comments are the most numerous, followed by posts, which is visible in the results shown in Table 1 where we can see their dominance in this regard: they hold 96% of the dataset weight. In order to decrease the coherence metric of the dataset, $CH(\mathcal{T}, D)$, we should decrease the coverage $CV(T, D)$ of each type T from the dataset. But, if we, for example, decrease the coverage of type *Person* from 0.95 to 0 (which is not realistic), that will result in a drop of the coherence measure for less than 0.3%. Bearing in mind that we have to decrease it much more than that, the only reasonable choice for modifications are the *Comment* and *Post* types.

The mutual predicates of these two types are *browserUsed*, *content*, *creationDate*, *hasCreator*, *hasTag*, *id*, *isLocatedIn*, *length* and *locationIP*, while *Comment* instances additionally have the *replyOf* predicate, and *Post* instances can have

language and *imageFile* properties if the *Post* instance is a photo. One way of decreasing the coverage of specific types is the removal of a high number of triplets related to a specific property. But, taking into account the specific domain, we conclude that the only property that can be removed in part of the posts and comments is *isLocatedIn*. The initial purpose of this property was to specify a country from which the message was issued, and it was determined by the IP address of the location where the message had been created. However, since a lot of users access social networks using their smartphones equipped with GPS receivers, social networks offer the possibility of adding locations to the messages. If we consider this property in that manner, we can remove the location from some messages, as not all messages contain location information. Various research in the domain show that users rarely share their location in the posts: the authors of [7] show that only about 1.2% of Twitter posts contain an explicit location information, while [9] shows that only around 6% of Instagram posts (photos) have a location tagged. Therefore, we remove the location information from 98% of comments and textual posts, and from 94% of photo posts, and with it the coverage of posts and comments gets significantly reduced.

Since it does not make sense to remove any other property, in order to achieve our goal, we decided to introduce new ones. In the initial dataset, all of the comments are textual, while recently social networks added a predefined set of GIFs which can be used as comments [8, 6]. In the initial dataset, one third of all comments are long textual comments, while two thirds are short comments, e.g. "ok", "great", "cool", "thx", "lol", "no way!", "I see", "maybe", etc. In order to include GIFs as comments, we introduce the *gifFile* property, which we apply in 80% of the short comments as a replacement of their *content* property.

In the next step, we add one more property to posts and comments: *mentions*. Its purpose is to mention a person in a post or a comment. This modification is also in line with what we have on social networks such as Facebook, Twitter, Instagram, etc., where a user can mention another user in a post or a comment, usually to make the other person aware of it. An analysis we performed over the Twitter7 dataset [13] showed that 40% of he tweets contain at least one mention. Therefore, we add this property to 40% of posts and comments, which provides an additional drop in the coherence measure.

A significant issue in operating a social network is privacy. Facebook introduced the possibility for each author of a post/comment to determine its level of privacy: if you want to share it publicly, to your friends only, or to specific group of people [1]. Therefore, we introduce the *visibility* predicate, which is set to a post/comment when it is posted with a privacy setting different from the default one for the user. Therefore, we generate this property for 5% of all messages, using the assumption that users generally use their default privacy setting.

The final change we added to the data generator is the addition of the *link* property, which both textual posts and comment can have. This corresponds to the real-world activity of sharing a link in a post, in addition to the text. Based on the analysis of user behavior on social media [10], which found that 43% of Facebook posts contain links, we add the *link* property to that share of textual

posts and comments. As a value for the *link* property, we use a random value from a predefined pool, similar as with other properties filled by the Datagen. It will always be fetched at the end of query execution, without any filtering to introduce estimation of cardinality, so the actual value is irrelevant.

The new SNB Datagen which generates RDF datasets with real-world coherence is publicly available on GitHub[3].

5 Measurements

To assess the structuredness of the RDF datasets generated by the original data generator and our modified version of it, we made measurements of the datasets in different sizes: 1, 3, 10, 30, and 100GB. Table 2 and Table 3 depict the results of the measurements: they show the number of triplets (in millions), and the coherence metric for all versions of the datasets. The tables provide a good overview of the dependence of the structuredness measure on the dataset size and the number of instances, in both the original and the modified dataset.

Table 2. Coherence of the initial SNB datasets.

SF	#Triplets	Coherence
1	46.9M	0.8599
3	142.6M	0.8702
10	480.8M	0.8808
30	1478.2M	0.8883
100	4804.3M	0.8943

Table 3. Coherence of the modified SNB datasets.

SF	#Triplets	Coherence
1	45.4M	0.6025
3	136.5M	0.6049
10	464.1M	0.6086
30	1428.4M	0.6115
100	4645.7M	0.6139

6 Conclusion and Future Work

In this paper, we introduced modification to the SNB data generator, to lower the generated RDF dataset coherence to a value of around 0.6, which corresponds better with real RDF datasets. We removed the location value in most posts and comments, and introduced new properties in the dataset: a GIF-type comment, user mentions and links in posts/comments, as well a level of visibility of a post. We used general characteristics of real social networks, such as Twitter, Facebook and Instagram, to generate a dataset which mimics real social networks. With all changes combined, we manage to get an RDF dataset for the SNB with the desired structuredness, i.e. coherence value. Additionally, we introduce a set of Virtuoso procedures which can be used for calculating the dataset coherence of

[3] `https://github.com/mirkospasic/ldbc_snb_datagen`

any RDF dataset stored in the quad store. With this, we simplify the process of coherence calculation introduced by the authors of [3].

As future work, we plan to reduce the coverage of other types besides posts and comments. We will also address the correlations in the newly added parts of the dataset. This will not change the overall structuredness, but the dataset will further correspond to real-world RDF data. The changes and additions introduced in the dataset will be implemented in the corresponding SNB queries.

References

1. When I post something, how do I choose who can see it? `https://www.facebook.com/help/120939471321735`. Accessed: 2016-06-29.
2. Renzo Angles, Peter Boncz, Josep Larriba-Pey, Irini Fundulaki, Thomas Neumann, Orri Erling, Peter Neubauer, Norbert Martinez-Bazan, Venelin Kotsev, and Ioan Toma. The Linked Data Benchmark Council: A Graph and RDF Industry Benchmarking Effort. *ACM SIGMOD Record*, 43(1):27–31, 2014.
3. Songyun Duan, Anastasios Kementsietsidis, Kavitha Srinivas, and Octavian Udrea. Apples and Oranges: A Comparison of RDF Benchmarks and Real RDF Datasets. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, pages 145–156. ACM, 2011.
4. Orri Erling, Alex Averbuch, Josep Larriba-Pey, Hassan Chafi, Andrey Gubichev, Arnau Prat, Minh-Duc Pham, and Peter Boncz. The LDBC Social Network Benchmark: Interactive Workload. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 619–630, New York, NY, USA, 2015. ACM.
5. Orri Erling and Ivan Mikhailov. *Virtuoso: RDF Support in a Native RDBMS*, pages 501–519. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
6. Kia Kokalitcheva. There's Now a Better Way to GIF on Twitter. `http://fortune.com/2016/02/17/twitter-gif-button-finally/`, 2016. Accessed: 2016-06-29.
7. Kalev Leetaru, Shaowen Wang, Guofeng Cao, Anand Padmanabhan, and Eric Shook. Mapping the Global Twitter Heartbeat: The Geography of Twitter. *First Monday*, 18(5), 2013.
8. Molly McHugh. You Can Finally, Actually, Really, Truly Post GIFs on Facebook. `http://www.wired.com/2015/05/real-gif-posting-on-facebook/`, 2015. Accessed: 2016-06-29.
9. Simply Measured. Quarterly Instagram Network Study (Q4 2014). 2014.
10. Amy Mitchell, Jocelyn Kiley, Jeffrey Gottfried, and Emily Guskin. The Role of News on Facebook. `http://www.journalism.org/2013/10/24/the-role-of-news-on-facebook/`, 2013. Accessed: 2016-06-29.
11. Minh-Duc Pham, Peter Boncz, and Orri Erling. S3G2: A Scalable Structure-Correlated Social Graph Generator. In *Technology Conference on Performance Evaluation and Benchmarking*, pages 156–172. Springer, 2012.
12. Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The Anatomy of the Facebook Social Graph. *arXiv preprint arXiv:1111.4503*, 2011.
13. Jaewon Yang and Jure Leskovec. Patterns of Temporal Variation in Online Media. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 177–186. ACM, 2011.