Understanding Billions of Triples with Usage Summaries

Shahan Khatchadourian and Mariano P. Consens

University of Toronto
shahan@cs.toronto.edu, consens@cs.toronto.edu

Abstract. Linked Data is a way to share and consume interlinked semantic web datasets. Usage summaries can help to understand the structure within and across interlinked datasets by partitioning entities based on how they are described, such as grouping entities that are instances of the same types and described with the same predicates. Because Linked Data is growing to billions of triples, scalable techniques for generating usage summaries are essential.
We extend our previous work, ExpLOD, by implementing a novel Hadoop-based technique for generating usage summaries of billions of triples. We analyze and compare usage summaries generated for the *entire* BTC 2010 and 2011 datasets. We generate usage summaries involving classes and predicates, and of recommended patterns, such as for inferencing and interlinking.

1 Introduction

In order to continue to promote the production and consumption of datasets from the Linked Open Data (LOD)[1] cloud, it is worthwhile to understand the descriptions each dataset provides and how they interact with other datasets. The challenge is that, as the size of semantic web datasets grows to billions of triples, the number of descriptions can grow exponentially, and so, scalable techniques need to be developed in order to understand the data.

Our previous work, ExpLOD [4], provides a mechanism to create comprehensive *usage summaries*, where an entity's *usage* is characterized as in [2]. Summaries help to understand how published entities are used together by succintly capturing how entities are *described*, such as how sets of classes and predicates are used in conjunction. When applied to datasets from the LOD cloud, usage summaries reveal *all* the unique and varied descriptions that occur *within* a dataset as well as *across* interlinked datasets. Understanding facilitates consumption of linked data. ExpLOD's implementation was bound to datasets that fit in main memory, limiting the size of datasets for which usage summaries could be generated. In this paper, we propose a novel, scalable mechanism to generate usage summaries of billions of Linked Data triples.

[1] http://linkeddata.org/

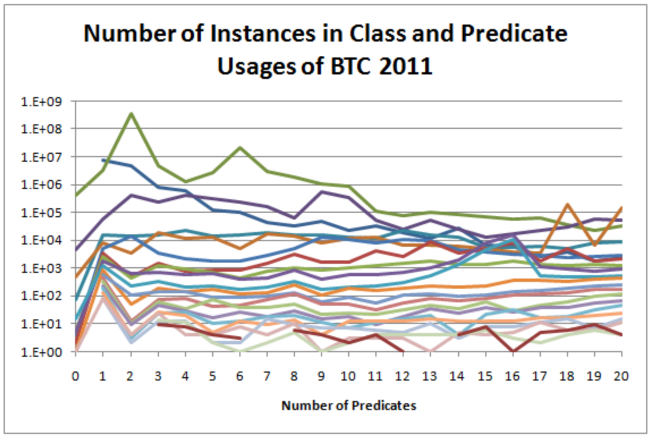


Fig. 1. Instances in class and predicate usage summary of BTC 2011

Related Work Analysis of the BTC dataset in previous years has generally relied on statistics based on popularity of single entities, such as the number of occurrences of a class based on the number of instances that it has been used to describe. Such statistics do not show how different classes and predicates interact and, as such, miss the opportunity for users to understand how descriptions vary both within and across different datasets.

Our novel approach differs from previous efforts in several ways. First, we extract *all* the unique descriptions and are not limited to a specific domain as in [5]. Second, our mechanism is flexible, allowing us to generate and compare many different usage summaries. In particular, we are not limited to portions of the dataset like in [1] - our work goes well-beyond previous efforts by processing and comparing the BTC 2010 and 2011 datasets in their *entirety*.

Contributions Our contributions in this work are as follows:

1. We describe how classes and predicates interact by generating and comparing numerous usage summaries of the BTC 2011 and BTC 2010 datasets.
2. Using several patterns described in [3] that are applicable to publishing and consuming Linked Data, such as for interlinking or inferencing, we generate usage summaries of these patterns and show how they are used in varied descriptions within the BTC datasets.
3. We describe our scalable, Hadoop-based implementation that can generate usage summaries over datasets containing billions of triples.

2 Generating Usage Summaries

ExpLOD's [4] usage summaries help a user's understanding of datasets from the Linked Open Data cloud. In this section, we introduce the reader to usage, usage neighbourhoods and usage summaries.

An entity's *usage* describes how it is used within a semantic web description, such as whether an entity is an instance, part of the schema (such as a class or predicate), or acts to create interlinks. A *usage neighbourhood* captures a set of usages that are used together in a semantic web description. For example, an instance that is typed as a *Person* has the *class* usage neighbourhood consisting of the singleton set {*Person*}. As another example, an instance that is a type of more than one class has a class usage neighbourhood that is the set of classes that it is a type of, such as {*Person, Organizer*}. Usage neighbourhoods can include multiple usages, such as class and predicate usages (what predicates are used to describe the instance) as a way to help understand the semantic web descriptions contained within the dataset and how the descriptions occur in conjunction.

Instead of exploring usage neighbourhoods of individual entities, which can be substantial if the dataset is sizeable, it is more convenient to create a *usage summary* that groups those entities that have the same usage neighbourhood. For example, a class and predicate usage summary groups those instances that have the same class usage and predicate usage. For example, two instances with class usage {*Person*} that have predicate usage {*homepage*} will be grouped together, but an instance with class usage {*Person, Organizer*} and predicate usage {*homepage*} will belong to a different group because it has a different class usage. Usage summaries are a succint way of exploring and understanding the *unique* semantic web descriptions of a dataset.

A usage summary is created by first constructing an RDF dataset graph from the input dataset, applying a *bisimulation label* to each node (described below), extracting relevant subgraphs that pertain to the usage neighbourhood of each instance in the dataset, then partitioning the instances based on their usage neighbourhood.

Usage summaries are generated from the graph representation of a semantic web dataset. A dataset's graph has directed edges and labeled nodes, and is constructed by having a node for each distinct URI that occurs as a subject or object in a triple, as well as a unique node for each statement's predicate. For each statement, an edge is drawn from the subject node to the predicate node, and from the predicate node to the object node (if it exists).

The bisimulation labels we use are based on a combination of the following: (1) Usage prefix: "C" for a class, "P" for a predicate. (2) Graph URI hostname: We consider only the main domain of a statement's graph URI. For example, we use `bestbuy.com` from the URI `http://products.semweb.bestbuy.com/products/16293707/semanticweb.rdf`. Techniques described in [1] can be used to ascribe instances to datasets. (3) Entity URI: We use the entity URI as is, e.g., `http://xmlns.com/foaf/0.1/homePage`. Bisimulation labels are applied to the dataset graph by examining subgraphs to determine each entity's usage. For example, the object of a statement with predicate *rdf:type* is used as a class, and the subject is an instance of that class, which generates a node labeled with the object URI and prefixed with "C", a node labeled "P+rdf:type" for the predicate, and a node with the instance URI.

	Label		
Summary	usage	graph	entity
complete	x	x	only if class or predicate
usage	x		
graph	x	x	
cpo	x		only if class or predicate
interlink	x	x	only if owl:sameAs, rdf:seeAlso, or skos:exactMatch
index	x	x	only if rdf:Seq or rdf:List
inference	x	x	only if skos:broader or skos:narrower
foaf	x	x	only if foaf:*
skos	x	x	only if skos:*
topic	x	x	only if like foaf:topic or foaf:PrimaryTopic

Table 1. Summaries considered and their bisimulation label

Table 1 shows the bisimulation labels for the usage summaries we have generated. We adopt a slight variation to the bisimulation label proposed in [4] by using “+” after the usage prefix, and “|” between the graph hostname and entity URI. The table specifies the composition of the bisimulation label for each summary type. For example, the *complete* bisimulation label concatenates the usage, the graph URI, and the entity URI, and the *cpo* bisimulation label concatenates the usage and the entity URI (which is either a class or predicate). An example bisimulation label for an entity used as a class in the *complete* usage summary is
’C+linkedmdb.org|http://data.linkedmdb.org/resource/movie/film’ and an example class bisimulation label in the *cpo* usage summary is ’C+http://data.linkedmdb.org/resource/movie/film’.

Technically, a usage summary is a partition of a set of subgraphs that represent usage neighbourhoods in the dataset graph. The subgraphs in each set are *bisimilar*, that is, the subgraphs in a set are pairwise “equivalent” and not with any subgraphs outside of their set. Specifically, two subgraphs are bisimilar if they each have a node with the same bisimulation label, and recursively, if the bisimulation label of nodes connected by outgoing edges are also the same. By employing multiple labeling schemes, we simplify the creation of different usage summaries. For example, when using the *cpo* bisimulation label the two entities described in Section 2 have the following class and predicate usage neighbourhoods:

- {C+Person, C+Organizer, P+rdf:type, P+homepage}
- {C+Person, C+Organizer, P+rdf:type}

3 Usage Summaries of Billions of Triples

In this section, we describe and compare the semi-structure of the 2010 and 2011 BTC datasets with usage summaries generated using the bisimulation labels listed in Table 1.

3.1 Class and Predicate Usage Neighbourhoods

We begin by analyzing the *cpo* usage summaries of the BTC 2010 and 2011 datasets. In the class and predicate usage neighbourhoods of BTC 2010 and BTC 2011 datasets, the 406,170,030 distinct instances in BTC 2010 are described with 3,281,294 distinct usage neighbourhoods. Meanwhile, the number of distinct instances in BTC 2011 has decreased to 390,358,846, and they are described with 2,790,334 unique class and predicate usage neighbourhoods.

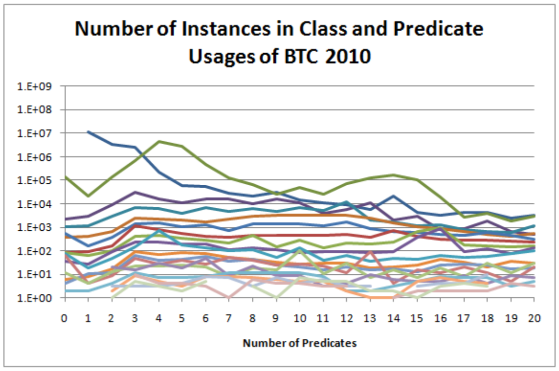


Fig. 2. Instances in class and predicate usage summary of BTC 2010

Figures 1 and 2 shows the number of instances in the unique class and predicate usage neighbourhoods of BTC 2010 and 2011, respectively. Each series line represents those neighbourhoods with a fixed number of classes. For example, the top-most line at the 1-predicate column has exactly 0 classes in its class usage, and the series lines underneath have fewer classes. The most popular class and predicate usage neighbourhood in both BTC 2010 and BTC 2011 (when including the graph hostname in the bisimulation label, as described in Section 2) is:

[C+hi5.com|foaf:Person, P+hi5.com|rdfs:seeAlso, P+hi5.com|foaf:nick], which indicates that the most commonly occurring description in the datasets are about instances described by the hi5.com hostname that are of type *foaf:Person*, are described only by *foaf:nick* predicates, and also have an interlinking predicate *rdfs:seeAlso*. In fact, 340,583,367 instances in the BTC 2011 dataset have this usage neighbourhood - this is visible in the peak at 2 predicates of Figure 1.

Figures 1 and 2 show that, for some usage neighbourhoods, as the number of classes increase, the number of instances decrease, and also that the number of instances decrease as the number of predicates increase. Visual comparison of the 2010 and 2011 datasets show similar trends, except that in 2011, the number of instances that are described with 1 or 2 predicates has increased. We also notice that the number of instances in BTC 2011 described with some usage neighbourhoods having around 9 or 16 predicates has increased in comparison to BTC 2010.

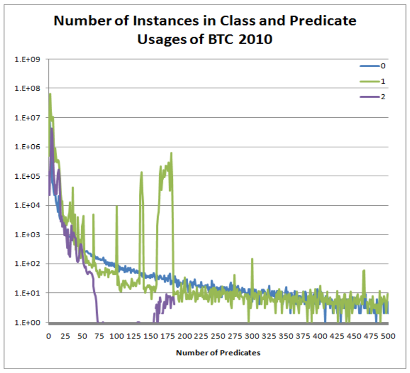


Fig. 3. Tail of BTC 2010

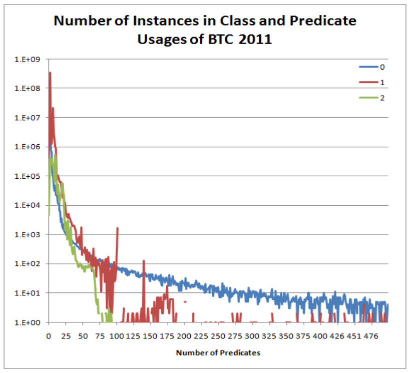


Fig. 4. Tail of BTC 2011

In Figures 1 and 2, notice that cpo usage neighbourhoods with 0 classes and 20 predicates describes thousands of instances. In the BTC 2011 dataset, only cpo usage neighbourhoods with 0 classes and at least 229 predicates describe fewer than 10 instances each. To show how the number of instances described with usage neighbourhoods having a high number of predicates varies, we show the number of instances in usage neighbourhoods with 0,1, and 2 classes with up to 500 predicates in Figures 3 and 4. We note that cpo usage neighbourhoods with 0 classes have a long tail, unlike cpo usage nieghbourhoods with at least 1 class. BTC 2010 also has a prominent number of instances that are described using 1 class and fewer than 200 predicates, a trend that does not appear in BTC 2011.

3.2 Usage Neighbourhoods of Linked Data Patterns

In this subsection, we explore patterns that have been described in [3]. Our work focuses on *structure-based* patterns rather than value-based equivalence, such as the equivalence of two literals "1" and "1.0". Instead, we. The bisimulation labels we use for the following summaries were shown in Table 1, and take into consideration the originating graph hostname of each usage.

The patterns we explore in this work are the following:

- Topic Relation, Composite Descriptions: where the predicate is like foaf:topic or foaf:primaryTopic.
- Link Base, See Also, Equivalence links: where the predicate is one of owl:sameAs, skos:exactMatch or rdfs:seeAlso
- Materialize inferences: where the predicate is like skos:broader and skos:narrower.
- Index Resources: where the instance is typed as a rdf:List or rdf:Seq
- We also examine how the FOAF and SKOS ontologies are used in descriptions in Section 3.3.

	Usage Neighbourhoods		Instances	
	2010	2011	2010	2011
Inference	120	256	68,672	842,938
Interlink	2,824	2,212	69,563,476	363,104,656
Topic	2,380	6,223	222,267	1,071,213
Index	12,324	60	710,972	769,138

Table 2. Pattern summaries of BTC 2010 and BTC 2011

Table 2 shows the number of distinct usage neighbourhoods and instances involved in the summaries described above compared between BTC 2010 and BTC 2011. For example, the first row of the table says that in the inference usage summary, BTC 2010 contains 68,672 instances divided amongst 120 distinct usage neighbourhoods, while BTC 2010 contains 842,938 instances divided amongst 256 distinct usage neighbourhoods, a substantial increase; topical descriptions also experience an increase in usage neighbourhoods and instances. The number of interlinked instances has increased by 5 times despite being expressed with fewer unique usage neighbourhoods. We also recognize that the number of indexed resources remains similar, but are expressed using much fewer unique usage neighbourhoods.

3.3 FOAF and SKOS Usage Neighbourhoods

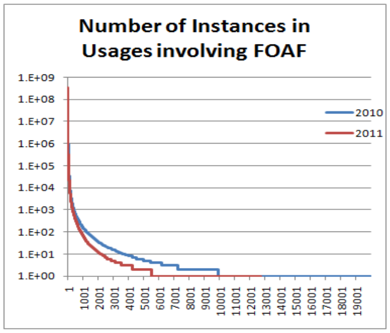


Fig. 5. FOAF schema usage

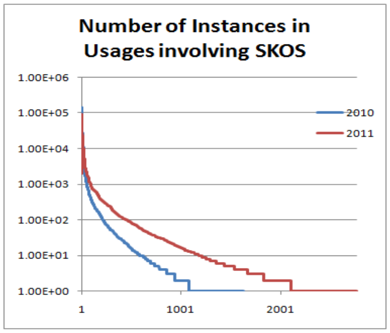


Fig. 6. SKOS schema usage

Figure 5 shows the number of instances in that usage neighbourhoods involving the FOAF ontology for the BTC 2010 and BTC 2011 datasets. It reveals that BTC 2011 has fewer distinct usage neighbourhoods as well as fewer instances than in BTC 2010. This could be as a result of a shutdowns of social networking sites such as vox.com, which was the eleventh-most common FOAF usage neighbourhood in BTC 2010, but removed from BTC 2011. Figure 6 shows SKOS usage in the BTC 2010 and BTC 2011 datasets, and reveals that the BTC 2011 dataset has increased the number of distinct usage neighbourhoods and describes more instances.

4 Implementation

We first generate the most detailed summary, the *cpo* summary specifically, which is based on the bisimulation label that concatenates the usage prefix, graph hostname, and entity URI. We then generate coarser summaries (that use include fewer details in their bisimulation label). Although we have a basic SPARQL-based implementation using Jena, we implented the summaries using low-level string- and line-based parsing. We did this for economic reasons - processing graphs takes at least an order of magnitude longer.

The *cpo* summary, from which all other summaries were generated, is computed using two Hadoop jobs. The first job takes a BTC dataset, GZipped NQuad files read using the NxParser library[2], as input, and computes the *cpo* usage neighbourhood for each distinct subject in the dataset; each unique blank node identifier that appeared as a statement's subject was also included. The job outputs GZipped tab-separated files containing the subject URI, and its *cpo* usage neighbourhood.

The second job takes the output of the first job and switches the key and value, so that the reduce task groups subjects by their usage neighbourhood (using string comparison). The output of this job is the *cpo* usage summary, which is stored as compressed tab-separated files containing each distinct usage neighbourhood and the number of instances with that usage neighbourhood. These two jobs completed on 40 Amazon Elastic MapReduce[3] large instances in under 40 minutes each. Since the *cpo* summary output is only 10% of the original dataset size, it becomes feasible to analyze the remainder of summaries locally. Other summaries are generated using a similar approach.

5 Online Exploration Tool

Figure 7 displays a screenshot of our online exploration tool accessible at **http://www.cs.toronto.edu/~shahan/swc2011/**. An interactive line graph can be moused-over to display the usage neighbourood at specific points, and selecting a point also selects its respective data row in the table appearing underneath the graph. The charts are implemented with the Google Charts API displaying data uploaded to Google Spreadsheets.

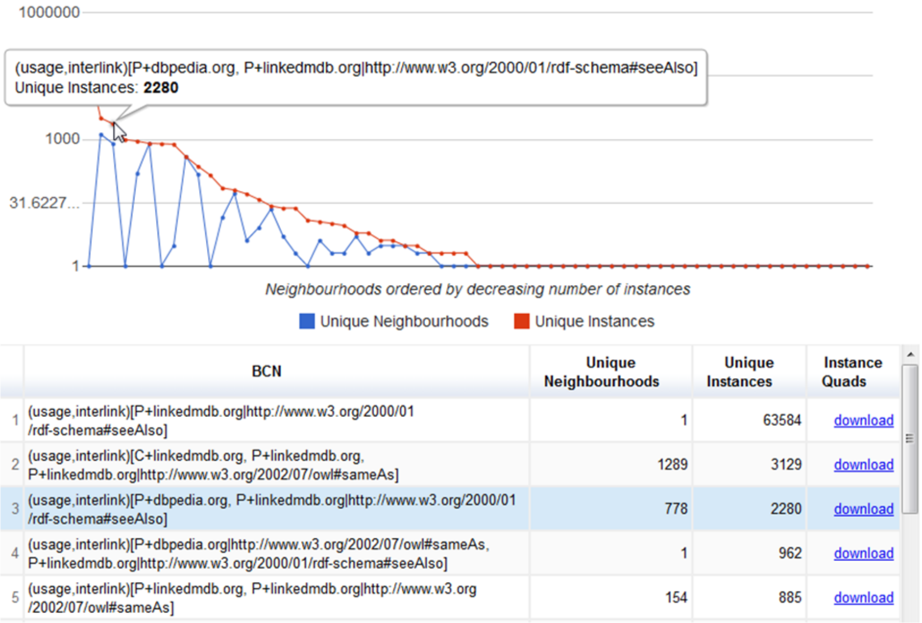
6 Conclusions

In this work, we have described a scalable technique to generate usage summaries of Linked Data containing billions of triples. Using flexible bisimulation labels, we generated and compared several usage summaries to aid understanding of how classes and predicates are used in the entire BTC datasets. Additionally, we have reported the variations of patterns that have been recommended such as for interlinking and inferencing.

Semantic Web Challenge, Billion Triple Track Requirements Our work, which uses the entire BTC 2010 and 2011 datasets, drives an analysis of Linked Data using increasingly more detailed usage summaries. Usage summaries that describe how instances, classes, and predicates are used together benefit Linked Data consumers by giving them insight to the descriptions contained in

[2] http://code.google.com/p/nxparser/

[3] http://aws.amazon.com/elasticmapreduce/



1000000
(usage,interlink)[P+dbpedia.org, P+linkedmdb.org|http://www.w3.org/2000/01/rdf-schema#seeAlso]
Unique Instances: **2280**
1000
31.6227...
1
Neighbourhoods ordered by decreasing number of instances
🟦 Unique Neighbourhoods
🟥 Unique Instances

	BCN	Unique Neighbourhoods	Unique Instances	Instance Quads
1	(usage,interlink)[P+linkedmdb.org|http://www.w3.org/2000/01/rdf-schema#seeAlso]	1	63584	download
2	(usage,interlink)[C+linkedmdb.org, P+linkedmdb.org, P+linkedmdb.org|http://www.w3.org/2002/07/owl#sameAs]	1289	3129	download
3	(usage,interlink)[P+dbpedia.org, P+linkedmdb.org|http://www.w3.org/2000/01/rdf-schema#seeAlso]	778	2280	download
4	(usage,interlink)[P+dbpedia.org|http://www.w3.org/2002/07/owl#sameAs, P+linkedmdb.org|http://www.w3.org/2000/01/rdf-schema#seeAlso]	1	962	download
5	(usage,interlink)[P+linkedmdb.org, P+linkedmdb.org|http://www.w3.org/2002/07/owl#sameAs]	154	885	download

Fig. 7. Screenshot of online exploration tool

the dataset. In addition, recommended publication patterns are also explored as usage, showing the ease and flexibility of our approach. Since our implementation is Hadoop-based, it is scalable; however, processing these datasets required only a few dollars worth of initial investment to generate the most detailed summary desired, after which coarser summaries were generated locally, each summary taking generally taking less than an hour to compute. We also generate usage summaries based on properties relevant to real-world semantic web applications, such as interlinking and inferencing.

References

1. C. Böhm, J. Lorey, D. Fenz, E. Kny, M. Pohl, and F. Naumann. Creating voiD descriptions. Semantic Web Challenge 2010.
2. L. Ding and T. Finin. Characterizing the semantic web on the web. In *ISWC*, pages 242–257, 2006.
3. L. Dodds and I. Davis. Linked data patterns. `http://patterns.dataincubator.org/book`, Sept. 2011.
4. S. Khatchadourian and M. P. Consens. ExpLOD: Summary-based exploration of interlinking and RDF usage in the linked open data cloud. In *ESWC (2)*, pages 272–287, 2010.
5. G. T. Williams, J. Weaver, M. Atre, and J. A. Hendler. Scalable reduction. Semantic Web Challenge 2009.