Statistical Analysis of Web of Data Usage

Markus Luczak-Rösch and Markus Bischoff

Freie Universität Berlin, Networked Information Systems WG, 14109 Berlin, Germany,
markus.luczak-roesch@fu-berlin.de, markus.wt@web.de
WWW home page: http://www.ag-nbi.de

Abstract. The Linked Data initiative gained momentum inside as well as outside of the research community. Thus, it is already an accepted research issue to investigate usage mining in the context of the Web of Data from various perspectives. We are currently working on an approach that applies such usage mining methods and analysis to support ontology and dataset maintenance tasks. This paper presents one part of this work, namely a method to detect errors or weaknesses within ontologies used for Linked Data population based on statistics and network visualizations. We contribute a detailed description of a log file preprocessing algorithm for Web of Data endpoints, a set of statistical measures that help to visualize different usage aspects, and an examplary analysis of one of the most prominent Linked Data set – DBpedia – aimed to show the feasibility and the potential of our approach.

Keywords: linked data, web usage mining, ontology maintenance

1 Introduction

The Linked Data initiative gained momentum inside as well as outside of the research community. At least the recent open government data approaches stress that assumption. That means that it is reasonable to expect that the real world usage of Linked Data, in the sense of querying and accessing it, will increase. It is already an accepted research issue to investigate usage mining in the context of the Web of Linked Data (or short: Web of Data). We are currently working on an approach that applies such usage mining methods and analysis to support dataset ontology maintenance. This paper presents one part of this work, namely a method to detect errors and weaknesses within ontologies used for Linked Data population based on statistical measures and their visualization by use of a network analysis tool.

1.1 Motivation, Terminology and Challenges

It is not in all cases trivial to apply the methods from classical Web usage mining to this new discipline one could call *Web of Data usage mining*. A first problem is the terminology as it is familiar for people in the context of the Web of documents. To our best knowledge only one W3C effort exists which aimed to define a terminology that characterizes the structure and the content of the Web[1]. This terminology does not cover the entities properly which are of interest on the Web of Data: *resources* that represent individual "things" named by URIs (or IRIs respectively) and a collection of *RDF statements* about such resources served in one place – a *dataset* – maintained by a *Web data publisher*. So far this is only a need for an adapted set of terms. But, even though it is not a requirement of a Linked Data endpoint to offer a SPARQL endpoint, lots of dataset providers on the Web of Data do so. Hence, resources on the Web of Data are requested directly via their URIs and by use of SPARQL queries which raises at least one central problem: The Web server observes requests for only one single Web resource very often (the SPARQL endpoint URI) while potentially more than one resource has been accessed as part of the query patterns.

Analyzing server logs is an intuitive way to perform Web usage mining. However, another problem on the Web of Data in its current shape is that the meaning of HTTP status codes[2] does not work out at all time. When accessing a URI which does not point to any resource on a Web server, the server responds the 404 code. The SPARQL protocol[3] requires servers to respond the 200 HTTP status code and a serialization of the SPARQL results format that contains no bindings in the case that a SELECT query is performed correctly but yields an empty result set. The HTTP 1.1 status code definitions[4] would recommend the use of the 204 status code in this case. This looks like a misuse of HTTP response codes at a first sight but also may be a desired feature for developers which deal with empty result sets application-dependent and detect this when the serialization of the result is processed. During our intensive work with logs from several Web of Data endpoints such as DBpedia[5], the Semantic Web Dog Food server[6], and Linked Geo Data[7] we observed that queries must be re-ran to find out whether they returned any result or not.

Listing 1.1. Anonymized excerpt of a DBpedia log file showing some of the different types of requests and the responded HTTP status codes.

```
xxx.xxx.xxx.xxx - - [21/Sep/2009:00:00:00 -0600]
  "GET /page/Jeroen_Simaeys HTTP/1.1"
  200 26777 ""
  "msnbot/2.0b (+http://search.msn.com/msnbot.htm)"
xxx.xxx.xxx.xxx - - [21/Sep/2009:00:00:00 -0600]
  "GET /resource/Guano_Apes HTTP/1.1"
  303 0 ""
  "Mozilla/5.0 (compatible; Googlebot/2.1;+http://www.google.com/bot.html)"
xxx.xxx.xxx.xxx - - [21/Sep/2009:00:00:01 -0600]
  "GET /sparql?query=PREFIX+rdfs%3A+%3Chttp%3A%2F%2Fwww.w3.org..."
  200 1844 ""
  ""
```

[1] http://www.w3.org/1999/05/WCA-terms

[2] http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

[3] http://www.w3.org/TR/rdf-sparql-protocol/

[4] http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

[5] http://dbpedia.org

[6] http://data.semanticweb.org/

[7] http://linkedgeodata.org/

The above mentioned problems show that it is an interesting issue to analyze usage on the Web of Data – especially requests against SPARQL endpoints. This paper deals with the research question *how usage analysis can support the maintenance of linked datasets*. Altogether we contribute three central things: First, an innovative log file preprocessing algorithm for Web of Data endpoints. Second, a set of statistical measures that help to visualize different usage aspects. Third, a statistical analysis of the usage of the DBpedia dataset with the purpose to identify problems in the data or the underlying schema. The remainder of this paper is structured as follows: Firstly we present a survey of related work in the following subsection. Afterwards Section 2 will introduce our preprocessing algorithm for log files of Linked Data endpoints before Section 3 describes the set of statistics and visualizations we propose for the analysis of the usage data. The Sections 4 and 5 complete this work with an evaluation of our approach by an examplary study and a discussion of the results as well as an outlook on future work.

1.2 Related Work

Classical Web usage mining has been placed within the Web mining hierarchy as a child of Web mining and a sibling to Web content mining [7]. Essential parts of Web usage mining are the characteristic metrics and patterns one has to identify, such as hits, page impressions, visits, time and navigation heuristics, unique visitors, clickthrough, viewtime, sessions, path analysis, association rules, sequential patterns, classification rules or clustering [13,14]. In this work we do not apply complex data mining methods to our data, such as sequential pattern mining or clustering, but remain on the statistical level.

We mentioned several differences between the classical Web and the Web of Data with reference to usage mining methods and techniques beforehand. Such a difference is also recognizable when we regard the use of the Web of Data in practice which has been described in works such as [6],[8] and [9]. Altogether, one can summarize that Linked Data typically is used (1) to provide unambiguous concept identifiers within Web applications, (2) to enhance the experience of Web users by aggregation and integration of corresponding content within CMS systems and Web applications, and (3) to be browsed and mashed up in a user-specific way. It becomes apparent that the classical browsing scenario plays a minor role and is outperformed by the access and use of Web resources through libraries or applications which are not or only indirectly connected with a human user's interaction and the SPARQL[8] query language plays an important role in these scenarios.

Already in 2002 and again in 2004 Berendt et al. [2,3] identified a new research area – the so called Semantic Web mining. The authors describe how the two disciplines, namely the Semantic Web and Web mining, may converge. They present three perspectives which reflect this: First, the perspective how Web mining can help to extract semantics from the Web. Second, the exploitation of

[8] `http://www.w3.org/TR/rdf-sparql-query/`

semantics for Web mining. And third, the perspective of mining of the Semantic Web. The latter perspective is the one which matches best to the focus of our work. It is subdivided into *Semantic Web structure and content mining* as well as *Semantic Web usage mining*. Again, the latter point is the one which is the most interesting one with reference to our work because it deals with the analysis of the usage of semantic data on the Web. Even though Berendt et al. mention one early approach that could result in log files which contain information about the usage of semantically rich content[10], it seems that since that date the research in that area and in the analysis of such log files was not very active.

Today this area gains a new momentum due to the broader success of the Linked Data ideas. To our best knowledge, in 2010 Möller et al.[12] published the next notable piece of work in this area. As a motivation for Linked Data usage analysis the authors raise a set of challenges, namely *reliability*, *peak-load*, *performance*, *usefulness*, and *attacks*. Möller et al. address these challenges by analyzing raw logs in order to learn about user clients, requested content types, and the structure of SPARQL queries. Our work will rely on the above mentioned challenges but address them under a different scope. We preprocess the logs in order to analyze the usage data on the level of basic graph patterns and the ontology primitives used in them.

Also after a very recent workshop on usage analysis and the Web of Data[9][4,5] this perspective is still unique. Only two papers at the workshop were related to log file analysis and worked upon the USEWOD challenge dataset which is partially a subset of the data we are working on. Kirchberg et al.[11] present an approach that combines data about real world events and log files to retrieve a notion of time-windowed relevance of data. Using an analysis of the syntactical and structural use of SPARQL in real-world scenarios to provide recommendations for index and store designers was introduced by Arias et al. [1].

2 Log File Preprocessing

To overcome the above mentioned issues with log files of Web of Data endpoints we propose an innovative preprocessing method. Our approach runs on server log files following the extended common log format[10]. These logs contain information about the access to RDF resources via their URIs and SPARQL queries. The first step of our preprocessing is to clean the log from all entries that contain 40x and 50x response codes. Afterwards we transform each single request for resources into a SPARQL DESCRIBE query to retrieve a normalized view to the usage of the dataset on the level of SPARQL queries. For all (1) basic graph patterns and (2) triple patterns of each single query, as well as the original query itself, we perform auto-generated queries that result in information about the success of individual graph patterns, triple patterns and the existence of resources and predicates in the dataset. The pseudocode of our algorithm is shown in Listing 1.2 and the resulting usage database in Figure 1.

[9] `http://data.semanticweb.org/usewod/2011/`

[10] `http://www.w3.org/TR/WD-logfile.html`

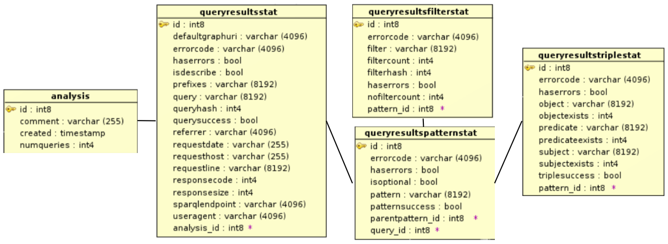


Fig. 1. Schema of the resulting database of the log file preprocessing

Listing 1.2. Pseudocode of the preprocessing algorithm

```
if(response_code < 400){
  if(request_string.isQuery()){
    query = request_string.extractQuery();
    query_suceeds = query.hasResults(endpointURI);
    foreach(graphpattern in query){
      pattern_query = "SELECT * WHERE " + graphpattern;
      pattern_succeeds = pattern_query.hasResults(endpointURI);
      if(graphpattern.isSubPattern()) storeParentPattern();
      if(graphpattern.isOptionalPattern()) isOptional = true;
      foreach(triplepattern in graphpattern ){
        triple_query = "SELECT * WHERE " + triplepattern;
        triple_exists = triple_query.hasResults(endpointURI);
        subject_query = "SELECT * WHERE
          {{ <"+subject+"> ?property ?hasValue }
          UNION { ?isValueOf ?property <"+subject+"> }} LIMIT 1 "
        subject_exists = subject_query.hasResults(endpointURI);
        predicate_query = "SELECT * WHERE
          {?s <"+predicate+"> ?o} LIMIT 1 "
        predicate_exists = predicate_query.hasResults(endpointURI);
        object_query = "SELECT * WHERE
          {{ <"+object+"> ?property ?hasValue }
          UNION { ?isValueOf ?property <"+object+"> }} LIMIT 1 "
        object_exists = object_query.hasResults(endpointURI);
        if(pattern.hasFilter()){
          resultSizeWithFilter();
          resultSizeWithouFilter();
        }
      }
    }
  else flag_and_run_as_describe();
}
```

3 Visualization of Web of Data Usage

The visualization of the collected data is done with an extension of the software “SONIVIS:Tool”[11] which enables network generation and analysis. We implemented network visualizations different perspectives on usage data, e.g. ontology, request hosts or time perspectives. Each perspective is supported by a set of widgets that represent detailed information about a selected entity of the network. To visualize the usage data on the basis of a given ontology, a transformation

[11] see http://sonivis.org

of the preprocessed data is necessary. Hence, a mapping between the resources used in queries and the classes which represent the corresponding types in an ontology which was used for data population in the respective dataset is established. In this section we introduce each of the implemented visualizations, the underlying metrics and interpretations of observations which are possible due to the visualizations. We do not present images of each visualization here due to limited space but we do so for a representative selection in Section4.

3.1 Ontology Heat Map

The ontology heat map provides an overview of the associated ontology primitives[12] of resources and predicates being used in queries. This is the global perspective on ontology usage. Its concept of a network visualization with weighted nodes and edges as a so called *heat map* is the basic concept of all further visualization as well.

Views: The central network view shows how often a specific primitive was used in queries. The more a certain primitive is used, the bigger the corresponding node in the graph view becomes and a specific color is applied to it. Zoom levels enable to focus parts of the network which are of a special interest. Two widgets contain lists that support (a) the examination of corresponding primitives of the resources that are present in the collected usage data and (b) statistical results for each primitive (count, absolute, relative).

Metrics: The view is based on metrics that sum the number of requests for each primitive that appears in triple patterns. “Count” is the absolute number of occurences used as a specific part of triple patterns. “Absolute” is the percentage of triple patterns using a chosen primitive out of all requested triple patterns. “Relative” is the percentage of queries that had no variable in the part of the triple pattern and used the chosen primitive.

Interpretation: With the results of this visualization one gets an overview of the general usage of an ontology which was used for data population. It is possible to see which ontology primitives are the most important ones for the users. For example resources of a specific class being used in 50% of the queries seems to be very important for the users while a class may be deleted from the ontology if no one accesses instances of it. The heat map is a starting point to analyze suspicious primitives in detail by use of other visualizations.

3.2 Primitive Usage Statistics

The primitive usage statistics help to find out in which context specific primitives were used, i.e. in which combination of classes and predicates.

[12] A primitive is a class if a subject or object in a triple pattern is analyzed. It is derived by resolving the rdf:type property of the resource. A primitive is a property when a predicate is analyzed.

Views: The visualization offers three perspectives: “Subject”, “Predicate” and “Object” each of them indicating the part of a triple for which a primitive is valid. In “Subject” for example one can choose a class that was used as subject and visualize as a graph which predicates or respectively objects are used in combination with it.

Metrics: The underlying metric groups the primitives being used in combination with a selected class/property, counts how often such a pattern was used and returns whether the triple succeeded.

Interpretation: With this view one can examine in which information users are interested in with respect to a specific class or property. For example if only one specific predicate is used in combination with a class. That means that users are only interested in one property of the class. With these information the Web data publisher can reason on how the ontology and the data should be evolved. If a combination is often and successfully used then the modelling of the ontology was well and there is data conforming to it. If another combination, which is conceptually possible, is queried very often but fails most of the time, it means that there is not enough data. It could be interesting to further investigate the triples of the query to get to know in which resources the users were interested in exactly. If a combination was used often but is not modelled in the ontology it could be necessary to adjust the ontology to enable this pattern if the queries are reasonable. Furthermore it is possible that some combinations of primitives that are modeled in the ontology but are never used in queries. In this case one could think about deleting this model and the according data.

3.3 Resource Usage Statistics

The resource usage statistics provide a more detailed view on triples that match a chosen pattern of primitives. Furthermore the view gives information if the ontology conforms to these triple combinations.

Views: The resource usage statistic is directly accessed from the primitive usage statistics and offers not a network visualization but a group of widgets. The core is a table containing all triples that match a chosen primitive combination. Below this table there are lists that contain the classes (or properties) associated to the resources of a chosen triple. If a predicate is focused in combination with other primitives there are two lists showing the domain and range of the predicate so that one can see if the ontology permits the observed usage. If the selected combination contains a subject-object pattern then here is a table that shows all properties that can be used between the two resources. A colored rectangle indicates whether the triple can conceptually be answered successfully (it is green) or if the ontology does not allow this combination of primitives (it is red).

Metrics: This metric aggregates every triple conforming to a selected triple pattern. It calculates how often each distinct triple was requested. To check if a triple request can conceptually be answered, the respective types of the resources as well as the domain/range of the property are compared to determine if the triple pattern is conceptually possible.

Interpretation: With this view one can get a close look on the resources the users are interested in and one can find the answer why a specific primitive combination failed. In general a request fails when there is no data that matches the query. This can have two reasons: (1) The ontology is modeled in a way that the combination is possible but there is a lack of data. (2) The successful answering of a triple is conceptually not possible, so there cannot be any valid data. In the first case one could extend the dataset as the users seem to be interested in these facts. In the second case and if a lot of users asked for such a failing combination one could decide if it is a good idea to extend the ontology and populate data. When data is detected that does not conform to the ontology, there are inconsistencies between the dataset and the ontology which should be examined as described in 3.6.

3.4 Hosts Statistics

The hosts statistics visualize the point of origin of requests as well as the request time of different hosts.

Views: The first view shows which classes a certain hosts uses in its queries as a table containing all hosts that request the dataset as well as the amount of requests of each host. Additionally it lists the classes and the number of the requests for it by the chosen host. The second view visualizes – starting from a class or property – which hosts used that primitive as certain part of a triple and how often this happened. The third view shows the request times of a selected host.

Metrics: The underlying metrics aggregate the distinct hosts which used the dataset and all triple patterns each host performed. Furthermore the request times of each host are calculated in an hourly format.

Interpretation: With the information provided by this perspective one can analyze the origin of requests and which parts of the dataset are used by different hosts. At first one gains statistical information about how many different hosts use the dataset, if there are hosts that make many requests at a certain point of time, or if different hosts access different and potentially specific parts of the data. Concretely, with the visualization of the primitives being used in queries of a specific host one can discover if a host has a regular set of patterns in the performed queries and a significant number of requests. Then the Web data publisher could serve the respective data separately to improve the performance of query answering for this host.

3.5 Time Statistics

The time statistics provide a global temporal view of the data and show how many requests were received by the service per hour to find out if there are times of high load.

Views: This view presents a bar chart that shows the number of all requests per hour. Below this chart there is a table containing the exact amount of queries for each time interval.

Metrics: The metric calculates the absolute number of queries in an hourly format.

Interpretation: As already mentioned, the amount of requests per hour can help to detect times of very high load. If that is the case one could decide to increase the server capacity so that the service does not break down during these times. On the other hand one can see times during which the dataset is not requested very often so the capacity of the server could be decreased to save resources and money. An observation of the time statistics over a time period can help to reason about the popularity of a dataset.

3.6 Error Statistics

The error statistics represent errors and missing things in the dataset to get information on what could be changed in the data and the underlying ontology to improve the dataset with respect to the users needs.

Views: The first view visualizes inconsistencies between the data and the ontology as a table of triples that should not exists conforming to the schema. Two additional lists contain the domain and range of the predicate so one can check which classes are permitted as subject and object. The second view shows combinations of classes and properties that are not modelled in the ontology which means that the predicate itself exists but that it cannot be used in combination with a certain class. The last view visualizes properties being used in requests but do not exist in the dataset. These properties and the amount of their usage are listed within a table.

Metrics: The first two metrics aggregate all requested combinations of primitives in triple patterns and checks in the ontology whether such a combination is allowed. For invalid combinations one metric checks whether there are requests for this pattern that succeeded which would be an inconsistency. The other metric simply lists the distinct triple patterns that are requested but fail due to the modeling. A third metric lists properties which are used in queries but which do not appear in the populated data.

Interpretation: With the inconsistency view one can easily see if there is data that should not exist. This data should be deleted or the ontology should be adapted to conform to it. The second view can provide information on how to modify or extend the ontology with respect to the users needs. If a lot of users request a specific property of a class it can be reasonable to modify the ontology and populate such data. With the third view one can observe which predicates are used that are not represented within the own dataset, for example properties of other ontologies or facts that have been deleted from the dataset. In the first case it could be a workaround to introduce "owl:sameAs" relations between the concept in the locally used ontology and the external one that models the same thing.

4 Evaluation

To evaluate our visualization concept for usage data derived from the preprocessed log files of Web of Data endpoints we ran the method experimentally on real world log data of the DBpedia 3.3 dataset. Therefor a local mirror of the DBpedia 3.3 dataset was set up for the preprocessing and the respective dbpedia 3.3 ontology was downloaded. We analyzed the log data of two randomly chosen days, namely 2009-07-02 and 2009-07-11. The number of requests which were analyzed was 631.512 and 1.083.390 respectively. This limited amount of days covered results from scalability issues of our method that requires a re-execution of queries, yet. We are aware that it would be reasonable to simply extend a SPARQL server library directly to produce the above mentioned usage data directly because this would avoid the effort of re-running each single query. However, the log file analysis respects the state of the art how servers on the Web of Data produce usage data. In the following we will present several exemplary visualizations which are the most significant ones for what we conclude from our analysis. It is not the goal of this paper to evaluate the usage of the DBpedia dataset completely but to prove the feasibility and the usefulness of our visual analysis approach in general. Thus, and due to limited space, we only present selected visualizations and corresponding interpretations which represent each of the possible maintenance recommendations our approach provides at least once. A broader extend of statistics and visualizations for both analysed log files can be found at http://page.mi.fu-berlin.de/mluczak/pub/visual-analysis-of-web-of-data-usage-dbpedia33/

Ontology Heat Map Analysis: For both datasets we used the ontology heat map as an indicator to step inside the DBpedia ontology and analyze specific primitives in detail. The visualizations depicted in Figure 2 indicate that only very few classes and properties have been used.

To proof that statistically for both datasets we generated the top 10 classes as deptictced in Table 1 and the usage of prroperties as shown in Table 2.

The most representative observations of the ontology heat map analysis are:

1. Requested resources are type of a small set of classes.

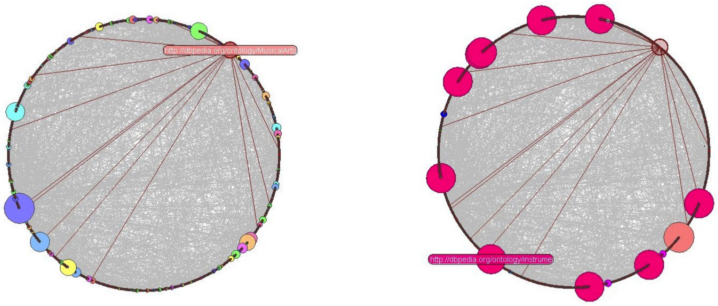

http://dbpedia.org/ontology/MusicalArtis
http://dbpedia.org/ontology/instrume

Fig. 2. Ontology heat map visualization for the log file from 2009-07-11 showing (a) the usage of classes and properties in an integrated graph and (b) the usage of properties used in queries.

2. Only very few properties are used as predicates in queries.
3. In general resources of more generic classes like Person, Organization and Place are used most frequently.
4. The class “MusicalArtist” is a hot spot in the log from 2009-07-11. Resources of this class were used about 19.000 times as a subject and 900 times as an object. The latter amount is more than 50% of all queries containing an object instead of a variable.
5. The data of 2009-07-11 indicates that there was a quite periodical and regular usage since there are some predicates which were requested exactly 1000 times, such as “associatedBand”, “instrument”, and “nationality”.

We conclude from observations 1, 2 and 3 that there is a potential to shape the ontology. Observation 4 and 5 indicate that it is reasonable to examine which other classes and properties are requested in context with “MusicalArtist” as well as “associatedBand”, “instrument”, and “nationality”. From 5 we also conclude that it is possible that there may be regular usage profiles of hosts.

Primitive and Resource Usage Analysis: With the observations of the heat map in mind the primitive and resource usage analysis as shown in Figure 3 helps to understand the users intention of requests better and to reveal concrete issues of the data and the underlying ontology.

Again we list the most important observations as follows:

1. A lot of triples containing a property like “birthdate” or “associatedBand” failed.
2. The predicates which are requested 1000 times occur almost everytime in combination with resources of the class “MusicalArtist”.
3. Most of the queries which used resources of the class “Band” failed since the ontology does not allow the requested combinations.
4. The property “instrument” is often used as an object in queries.
5. A lot of users query for the president of a certain country.

(a) 2009-07-02

Quantity	Subject			Object		
	Quantity	abs	rel	Quantity	abs	rel
Organization	69175	10.3%	19.1%	46984	7.0%	72.2%
Person	61845	9.2%	17.1%	187	<0.1%	0.3%
Place	25590	3.8%	7.1%	231	<0.1%	0.4%
Work	21394	3.2%	5.9%	34	<0.1%	<0.1%
PopulatedPlace	20455	3.0%	5.7%	221	<0.1%	0.3%
Company	19006	2.8%	5.2%	9688	1.4%	14.9%
Artist	17951	2.7%	5.9%	106	<0.1%	0.2%
Athlete	12530	1.9%	3.5%	0	0%	0%
EducationalInstitution	12288	1.8%	3.4%	9967	1.5%	15.5%
RadioStation	10968	1.6%	3.0%	9805	1.5%	15.1%

(b) 2009-07-11

Class	Subject			Object		
	Quantity	abs	rel	Quantity	abs	rel
Person	72458	6.9%	14.1%	141	<0.1%	8.2%
Artist	30526	2.9%	6.0%	81	<0.1%	4.7%
Place	27916	2.7%	5.5%	240	<0.1%	14.0%
Organization	22461	2.2%	4.4%	45	<0.1%	2.6%
Work	20767	2.0%	4.1%	15	<0.1%	0.9%
PopulatedPlace	20764	2.0%	4.1%	239	<0.1%	13.9%
MusicalArtist	18994	1.8%	3.7%	911	0.1%	53.0%
Athlete	13925	1.3%	2.7%	0	0%	0%
Actor	7708	0.7%	1.5%	30	<0.1%	1.7%
Company	7345	0.7%	1.4%	36	<0.1%	2.1%

Table 1. The top 10 of used classes

We conclude from observation 1 and 3 that the two examplary properties as well as the class "Band" are used in a different context than the one they are valid for. In the case of observation 1 it is also possible that there is a lack of data which conforms to the requests. 4 indicates that the identifier "instrument" is badly chosen or the users' understanding of this concept is different. A workaround could be to to change it to "playsInstrument" which reflects the character of a property more intuitively. It is modeled that a "Person" is "president" of a "school" but observation 5 revealed a reasonable query so the ontology could be adjusted to fulfill this user requirement. The observation 2 stresses the aforementioned assumption that there may be regular usage profiles of hosts which are worth of a detailed inspection.

Hosts and Time Analysis: The hosts and time analysis helps to detect hosts which only use a specific set of patterns, hosts which access the dataset at specific point of time, and times of heavy load in general. We discovered that the queries containing the properties which were used 1000 times originated from the same host. Figure 4 compares the access time of this host with the overall usage activity of all hosts. On both analyzed log files there was a constant traffic with about 30.000 (2009-07-02) and respectively 50.000 (2009-07-11) queries per hour.

The two central observations of this analysis are:

1. The load has an average distribution over the whole day.
2. There is at least one host that requests the dataset in a dedicated time period.

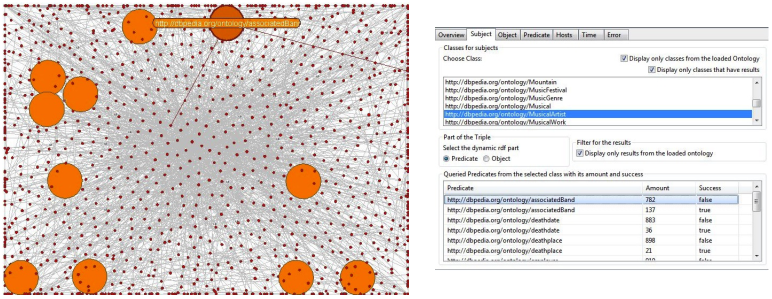
(a) 2009-07-02

Predicate	Quantity	abs	rel
birthdate	103	0.02%	0.04%
deathdate	72	0.01%	0.03%
birthplace	44	<0.01%	0.02%
knownFor	13	<0.01%	<0.01%
president	2	<0.01%	<0.01%
capital	1	<0.01%	<0.01%

(b) 2009-07-11

Predicate	Quantity	abs	rel
deathdate	1034	0.1%	2.7%
associatedBand	1000	0.1%	2.6%
deathplace	1000	0.1%	2.6%
employer	1000	0.1%	2.6%
genre	1000	0.1%	2.6%
instrument	1000	0.1%	2.6%
knownFor	1000	0.1%	2.6%
nationality	1000	0.1%	2.6%
occupation	1000	0.1%	2.6%
spouse	1000	0.1%	2.6%
term	1000	0.1%	2.6%

Table 2. The top list of used properties



Classes for subjects

Choose Class:

☑ Display only classes from the loaded Ontology

☑ Display only classes that have results

http://dbpedia.org/ontology/Mountain
http://dbpedia.org/ontology/MusicFestival
http://dbpedia.org/ontology/MusicGenre
http://dbpedia.org/ontology/Musical
http://dbpedia.org/ontology/MusicalArtist
http://dbpedia.org/ontology/MusicalWork

Part of the Triple

Select the dynamic rdf part

● Predicate ○ Object

Filter for the results

☑ Display only results from the loaded ontology

Queried Predicates from the selected class with its amount and success

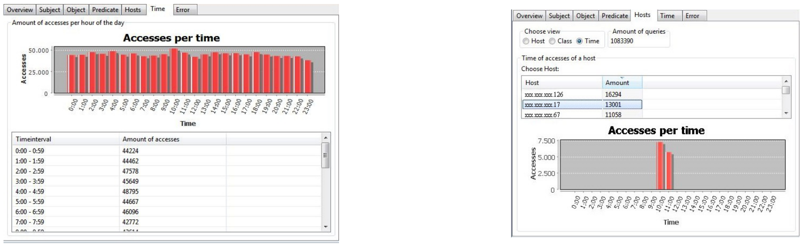
Predicate	Amount	Success
http://dbpedia.org/ontology/associatedBand	782	false
http://dbpedia.org/ontology/associatedBand	137	true
http://dbpedia.org/ontology/deathdate	883	false
http://dbpedia.org/ontology/deathdate	36	true
http://dbpedia.org/ontology/deathplace	898	false
http://dbpedia.org/ontology/deathplace	21	true

Fig. 3. Predicates which were used in combination with subjects of type MusicalArtist as (a) graph visualization and (b) detailed table view. (log from 2009-07-11)

Observation 2 allows at least the assumption that this dedicated host always requested the same query pattern as part of an experiment on the DBpedia dataset. But, the overall amount is still too limited to recommend to scale the infrastructure to improve the performance of the service or to modularize the data and serve modules separately for request hosts which need only a specific part of the dataset.

Error Analysis: Figure 5 depicts how we addressed the detection of inconsitencies in the data and the ontology. The scope of the exploration is defined by one of the primitives which was suspicious in the primitive and resource usage analysis such as “instrument” as well as “Band”.

1. There are triples in the dataset that use the property “associatedBand” but do not use a resource of type “MusicalArtist” as the respective subject which contradicts the domain restriction of “associatedBand”.
2. The users needs and the modeling of the ontology obviously differ with respect to the class “Band” since several predicates are requested in combination with resources of this type which are not valid.



Amount of accesses per hour of the day

Timeinterval	Amount of accesses
0:00 - 0:59	44224
1:00 - 1:59	44462
2:00 - 2:59	47578
3:00 - 3:59	45649
4:00 - 4:59	48795
5:00 - 5:59	44667
6:00 - 6:59	46096
7:00 - 7:59	42772

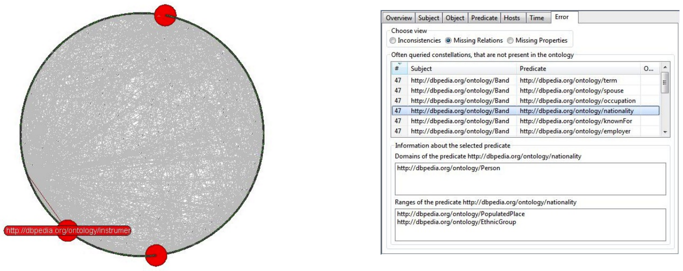
Choose view
○ Host ○ Class ● Time

Amount of queries
1083390

Time of accesses of a host
Choose Host:

Host	Amount
xxx.xxx.xxx.126	16294
xxx.xxx.xxx.17	13001
xxx.xxx.xxx.67	11058

Fig. 4. Access time statistics of (a) all requesting hosts and (b) one specific host. (log from 2009-07-11)



Choose view
○ Inconsistencies ● Missing Relations ○ Missing Properties

Often queried constellations, that are not present in the ontology

#	Subject	Predicate	O...
47	http://dbpedia.org/ontology/Band	http://dbpedia.org/ontology/term	
47	http://dbpedia.org/ontology/Band	http://dbpedia.org/ontology/spouse	
47	http://dbpedia.org/ontology/Band	http://dbpedia.org/ontology/occupation	
47	http://dbpedia.org/ontology/Band	http://dbpedia.org/ontology/nationality	
47	http://dbpedia.org/ontology/Band	http://dbpedia.org/ontology/knownFor	
47	http://dbpedia.org/ontology/Band	http://dbpedia.org/ontology/employer	

Information about the selected predicate
Domains of the predicate http://dbpedia.org/ontology/nationality
http://dbpedia.org/ontology/Person

Ranges of the predicate http://dbpedia.org/ontology/nationality
http://dbpedia.org/ontology/PopulatedPlace
http://dbpedia.org/ontology/EthnicGroup

Fig. 5. Error analysis for the log from 2009-07-11: (a) visualization of predicates with inconsistent data in the dataset and (b) requested triple patterns that are not valid with respect to the current ontology.

Our observation 1 reveals (a) an inconsistency in the populated data which potentially results from (b) errors in or the misuse of mappings for the Wikipedia infoboxes. The workaround could be to either to change the modeling of the domain restriction of "associatedBand" or to fix the mappings that they do not match the cases that result the invalid data. A detailed analysis for 2 shows that resources of the type "Band" are requested as subject in combination with the predicate "nationality" for example. These triples fail since "nationality" has a domain restriction to "Person". This could be fixed to represent the fact that e.g. a British band exist.

4.1 Summary

We summarize that we analyzed the general usage of the DBpedia dataset, detected inconsistencies in the populated data, and revealed requirements of users which are not modeled in the DBpedia ontology yet. With reference to the challenges introduced by Möller et al.[12] we discovered the following: The *error statistics* help to find out inconsistencies within served data and to improve the data quality which is an advance for the **reliability** of a dataset. The *time statistics* give an insight on times of heavy load. It is possible to guide from the time statistics, via the hosts statistics directly to the *primitives usage statistics* and the *resource usage statistics* which enable a detection of the most frequently requested entities of a dataset. So, both perspectives of **peak-load** are covered. The *hosts statistics* can be used to detect hosts which use a specific set of query patterns. The data conforming to these patterns could be served separately as a module of the entire dataset. Specific indexes covering the most important patterns could be configured based on the *ontology heat map*, the *primitives usage statistics*, and the *resource usage statistics*. Both activities can improve the system **performance**. The *ontology heat map* is an indicator for the suitability and conciseness of an ontology and the *error statistics* reveal requirements of the users which are not conform to it. Bringing both perspectives together one can draw conclusions about the **usefulness** of existing data as well as its modeling and restrain or extend the schema. The *time statistics* and the *hosts statistics* allow the detection of external **attacks**.

5 Discussion and Concluding Remarks

In this paper we presented an approach that helps Web data publishers to visualize and interpret the usage of their Web of Data endpoints with the goal to recommend maintenance activities such as the assurance of the systems performance or the fixing of bugs and weaknesses in the data or the underlying schema. We clearly motivated that such data analysis must be based on a specific preprocessing of log files and we proposed an algorithm for this. Then we presented six metrics, the associated visualizations, and the descriptions how they should be interpreted. The approach was evaluated by an exemplary usage analysis of the DBpedia dataset. The results of this analysis prove that our approach address a set of five accepted challenges properly. However, it does not seem to be reasonable to take all the derived maintenance recommendations into account for future evolution steps of the DBpedia dataset, yet. For example round about 1.500.000 people access the German Wikipedia per hour[13], which shows that the real-world usage of DBpedia is rather limited. Hence, we have to admit that in case of a broader public success of DBpedia such an analysis has to be re-performed.

The metrics and the associated visualizations which were presented in this paper are only a subset of all possibilities to perform a detailed usage analysis. Even though, the evaluation has shown that they are significant to reveal inconsistencies and weaknesses of a dataset and its underlying ontology. Also the most related piece of work by Möller et al.[12] only presented another subset of patterns and metrics for Web of Data usage analysis. Bringing both approaches together seems to be promising to get a complete view on the usage of Web of Data.

We are currently running further data mining experiments and analysis on our usage data. Hence, we are going to extend the statistical view to Web of

[13] Zachte, E., “Wikipedia Statistics - Europe, Friday December 31, 2010”, http://stats.wikimedia.org/EN Europe/Sitemap.htm, visited on February 2nd, 2011

Data usage with a content based view by application of cluster, session and path analysis. Furthermore we are currently using the preprocessed usage data for an approach that automatically adapts indexes of an RDF store based on the popularity and complexity of patterns in queries performed in real. The application of network visualizations in our approach also offers the chance to apply various network metrics (e.g. connectivity or centrality measures) to the ontologies. We are also going to experiment with these structural properties of various ontologies and the effects of changes on them which are concluded from our usage analysis.

References

1. Arias, M., Fernández, J.D., Martínez-Prieto, M.A., de la Fuente, P.: An empirical study of real-world SPARQL queries. CoRR abs/1103.5043 (2011), http://arxiv.org/abs/1103.5043
2. Berendt, B., Hotho, A., Stumme, G.: Towards semantic web mining. In: In International Semantic Web Conference (ISWC. pp. 264–278. Springer (2002)
3. Berendt, B., et al.: A roadmap for web mining: From web to semantic web. In: Berendt, B., et al. (eds.) Web Mining: From Web to Semantic Web. pp. 1–22. Springer, Heidelberg
4. Berendt, B., et al.: Usewod2011: 1st international workshop on usage analysis and the web of data. In: Srinivasan, S., et al. (eds.) WWW (Companion Volume). pp. 305–306. ACM
5. Berendt, B., et al.: Usage analysis and the web of data. ACM SIGIR Forum 45(11), 63–70 (2011)
6. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
7. Cooley, R., Mobasher, B., Srivastava, J.: Web mining: Information and pattern discovery on the world wide web. In: ICTAI '97: Proceedings of the 9th International Conference on Tools with Artificial Intelligence. p. 558. IEEE Computer Society, Washington, DC, USA
8. Hausenblas, M.: Exploiting Linked Data For Building Web Applications. IEEE Internet Computing 13(4), 68–73 (2009)
9. Heath, T.: How will we interact with the web of data? IEEE Internet Computing 12(5), 88–91 (2008)
10. Hotho, A., Maedche, A., Staab, S., Studer, R.: Seal-ii -the soft spot between richly structured and unstructured knowledge. Journal of Universal Computer Science (2001)
11. Kirchberg, M., Ko, R.K.L., Lee, B.S.: From linked data to relevant data–time is the essence. CoRR abs/1103.5046 (2011), http://arxiv.org/abs/1103.5046
12. Möller, K., Hausenblas, M., Cyganiak, R., Grimnes, G.A.: Learning from linked open data usage: Patterns & metrics. In: Proceedings of the WebSci10: Extending the Frontiers of Society On-Line
13. Spiliopoulou, M.: Web Usage Mining for Web Site Evaluation. Communications of the ACM 43(8), 127–134 (August 2000)
14. Srivastava, J., Cooley, R.: Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations 1, 12–23 (2000)