A Systematic Investigation of Explicit and Implicit Schema Information on the Linked Open Data Cloud

Thomas Gottron, Malte Knauf, Stefan Scheglmann, Ansgar Scherp

WeST – Institute for Web Science and Technologies
University of Koblenz-Landau
56070 Koblenz, Germany
{gottron,mknauf,schegi,scherp}@uni-koblenz.de

Abstract Schema information about resources in the Linked Open Data (LOD) cloud can be provided in a twofold way: it can be explicitly defined by attaching RDF types to the resources. Or it is provided implicitly via the definition of the resources' properties. In this paper, we present a method and metrics to analyse the information theoretic properties and the correlation between the two manifestations of schema information. Furthermore, we actually perform such an analysis on large-scale linked data sets. To this end, we have extracted schema information regarding the types and properties defined in the data set segments provided for the Billion Triples Challenge 2012. We have conducted an in depth analysis and have computed various entropy measures as well as the mutual information encoded in the two types of schema information. Our analysis provides insights into the information encoded in the different schema characteristics. Two major findings are that implicit schema information is far more discriminative and that applications involving schema information based on either types or properties alone will only capture between 63.5% and 88.1% of the schema information contained in the data. Based on these observations, we derive conclusions about the design of future schemas for LOD as well as potential application scenarios.

1 Introduction

Schema information of semantic data on the Linked Open Data (LOD) cloud is given in a twofold way: explicitly by providing the type of a resource and implicitly via the definition of its properties. These two manifestations of schema information are to a certain extent redundant, i.e., certain resource types entail typical properties and certain properties occur mainly in the context of particular types. For instance, we would expect a resource of type foaf:Person to have the properties foaf:name or foaf:age. Likewise, we can assume a resource with the property skos:prefLabel to be of type skos:Concept.

Schema information over LOD is used for various purposes such as indexing distributed data sources [10], searching in large graph databases [13], optimizing the execution of queries [14] or recommending appropriate vocabularies to linked data engineers [16]. Thus, it is an important question to which degree explicit and implicit schema information is correlated, i.e., to which extend the use of RDF types and properties appear together to describe resources. A high correlation of explicit and implicit schema information corresponds to redundant information—a fact which can be exploited, for instance, when indexing the LOD cloud and providing a central lookup table for LOD sources. One application in this context is the opportunity to compress a schema based index for LOD as motivated and requested by Neumann and Weikum [15]. More even, it is of interest, which schema information actually needs to be extracted from the Linked Open Data cloud and which information might be inferred[1]. Finally, a high correlation can be exploited directly for recommending typical combinations of types or properties when modelling Linked Data [16]. This leads us to the overall question to which extent the explicit schema information provided by RDF types coincides with the implicit schema information of the properties used in the LOD cloud and how consistent are the observed patterns and redundancies.

A fundamental prerequisite to answer this question is the availability of a reliable schema extracted from the LOD cloud that takes into account both explicit and implicit schema information. With the SchemEX approach [10,9], we can compute such a schema for huge amounts of RDF triples in an efficient manner. In the context of this paper, we describe a method and metrics for leveraging a schema obtained in this way to investigate the information theoretic properties and global dependencies between RDF types and properties. As the discussion of the related work in the subsequent section shows, such methods are—to the best of our knowledge—not available and an investigation as presented in this paper has not been done before. We will close this gap and consider for our analysis different data sets crawled from the LOD cloud and contained in the Billion Triples Challenge 2012 data set. The data sets cover data of different origin and quality and serve as basis for our experiments.

In Section 3, we will introduce a probabilistic schema distribution model. Based on this model, we identify different information theoretic metrics that are of interest. The metrics comprise different types of entropy as well as mutual information. In Section 4, we describe a method of how to estimate the relevant probabilities from a schema-based index and introduce the data sets we use for our analysis of linked data. The results of our investigation are shown in Section 5 where we also draw some conclusions regarding the design and application of future LOD schema. In summary, we have observed that the redundancy of explicit and implicit schema information on different parts of the LOD varied from 63.5% to 88.1%. Thus, a general schema for LOD should not be build on either explicit or implicit schema information only and should ideally integrate both types of information. Nevertheless, we also observed several highly indicative sets of properties, allowing a prediction of the types of resources.

2 Related Work

One application where schema information can be of value is query optimization. Neumann and Moerkotte [14] employ so-called *characteristic sets*, which basically classify RDF resources by the correlation of their (outgoing) predicate links. Knowledge about these sets allows for quite precise estimates of the result cardinality of join operations. Further insights into the correlation between properties in an RDF graph were not nec-

[1] Inference here can be realized in both ways: semantically or statistically.

essary. Neither were explicit schema information provided in form of RDF types considered. A similar approach is presented by Maduko et al. [13]. Here the focus was on efficient approaches to estimate subgraph frequencies in a graph database. This subgraph frequency information is then used for conducting efficient queries on the graph database. In their work, Maduko et al. use both implicit schema information and explicit schema information. However, they do not determine the cardinality of intermediate join results of the two schema information sources for executing the queries. Harth et al. [6] propose an approximative approach to optimize queries over multiple distributed LOD sources. They build a QTree index structure over the sources, which is used to determine the contribution of the single sources to the query results.

Several tools aim at providing statistics for the LOD cloud. LODStats [2] is a tool and framework for computing 32 different statistics on Linked Open Data such as those covered by the Vocabulary of Interlinked Data sets (VoID) [1]. The tool provides descriptive statistics such as the frequencies of property usage and datatype usages, the average length of literals, or counting the number of namespaces appearing at the subject URI position [2]. LODStats operates on single triple patterns, i.e., it does not provide statistics of, e.g., star patterns or other (arbitrary) graph patterns. However, it covers more complex schema-level characteristics like the RDFS subclass hierarchy depth [2]. Overall, analysis of the correlating use of different properties, RDF types, or the common appearance of properties and types like we investigate is out of scope. Also make-void[2] computes VoID-statistics for a given RDF file. These statistics usually contain information about the total number of triples, classes, properties, instances for each class, the uses of each property and the number of triples that link a subject on one domain to an object on another domain. Another framework for statistic generation on RDF data is RDFStats[3]. In contrast to make-void, RDFStats can also operate on SPARQL endpoints and uses a different vocabulary for its statistics.

Hogan et al. have conducted an empirical study to investigate the conformance of linked data sources with 14 different linked data principles [8]. As metric, the authors apply the number of unique namespaces used by the respective data providers and provide a ranked list in terms of top-5 and bottom-5 data providers. Among others, the authors analysed how different classes and properties of vocabularies defined at one data source are re-used and mixed by other linked data providers. In contrast, the analysis of the correlation of class terms and property terms of different (or the same) vocabularies done here is agnostic to the actual source the linked data originates from. Bizer et al. have recently analysed the joined occurrence of a single class with a single property on the structured data extracted from a large web crawl[4]. Lorey et al. [11] developed a frequent item set approach over properties for the purpose of detecting appropriate and diverging use of ontologies. None of these works addresses information theory metrics as it is done in the paper at hand. The application of information theoretic measures on RDF data is addressed in [12]. However, the analysis there is focussing on a different level of schema re-use of concepts and does not consider any property information.

[2] `https://github.com/cygri/make-void` (accessed 9 March 2013)

[3] `http://rdfstats.sourceforge.net/` (accessed 9 March 2013)

[4] `http://webdatacommons.org/` (accessed 9 March 2013)

3 Probabilistic Schema Model and Metrics

Schema information on the LOD cloud can be provided explicitly by the use of RDF type properties. There are no (practical) boundaries to the number of types that can be attached to a resource. In practice, we can observe resources which have no type as well as resources with several hundred types. In addition, schema information can be provided implicitly by the properties used to describe a resource. These properties connect one resource to another resource or a literal value. In this way, they implicitly describe the type of a resource by its relations. Again, it is possible to observe resources which have no relation (beyond a type description) as well as resources with hundreds of properties.

The goal of the analysis in this paper is to measure and quantify the information theoretic properties of the explicit schema information given by RDF types and the implicit schema information provided by the used properties. To this end, in Section 3.1 we first introduce a probabilistic model for the occurrence of types and properties of resources. This allows us to measure the schema information contained in types, properties or both together. In order to do so, we present different metrics such as entropy of marginal distributions, conditional entropy and mutual information in Section 3.2.

3.1 A Probabilistic Distribution Model for Types and Properties

We are interested in two observations about the resources on the LOD cloud: their types and their properties. To be more specific, we are interested in combinations of types and combinations of properties. A particular combination of types is a set of types attached to a resource. The space of all possible combinations therefore is the power set $\mathcal{P}(\mathit{Classes})$ of all class types in the data. While the power set itself is a huge set, we can actually restrict ourself to the subset $TS \subset \mathcal{P}(\mathit{Classes})$ of actually observed combinations of RDF types in the LOD cloud. For a given resource, we can now observe $t \in TS$ which corresponds to a set of types (e.g., the set $\{\mathsf{foaf{:}Person}, \mathsf{dbpedia{:}Politician}\}$).

Likewise, the properties observed for a resource is a combination of all possible properties. Accordingly here we deal with an element from the power set $\mathcal{P}(\mathit{Properties})$ of all observed properties. Again, we only need to consider the subset PS of actually occurred property sets. For an individual resource, we observe $r \in PS$ which corresponds to the set of its properties[5] (e.g., the set $\{\mathsf{foaf{:}familyName}, \mathsf{foaf{:}givenName}, \mathsf{dbpedia{:}spouse}\}$).

To model the joint distribution of type sets and property sets, we introduce two random variables T and R. These take as values the elements in TS and PS, respectively. Both random variables are of discrete nature and their joint distribution can be characterized by:

$$P(T = t, R = r) = p(t, r) \tag{1}$$

where $p(t, r)$ is the probability for a randomly chosen resource to observe the concrete set t of attached types and the set r of properties. Based on this joint distribution, we can also identify the marginal distributions of T and R:

[5] Please note, we use the letter r for sets of properties (inspired by the term relation), as p will be used to denote probabilities.

$$P(T=t) = \sum_{r \in PS} p(t,r) \quad , \quad P(R=r) = \sum_{t \in TS} p(t,r) \tag{2}$$

3.2 Metrics of Interest

For analysing the LOD cloud, we are interested in several characteristics of the joint distribution $P(T, R)$ introduced above. The main questions that we want to answer are:

(a) How much information is encoded in the type set or property set of a resource on a global scale?
(b) How much information is still contained in the properties, once we know the types of a resource?
(c) How much information is still contained in the types, once we know the properties of a resource?
(d) To which degree can one information (either properties or types) explain the respective other?

To answer these questions, we introduce appropriate metrics that can be applied to the joint distribution of type sets and property sets. All our metrics are based on the *entropy* of probabilistic distributions [17], the standard concept to measure information.

Entropy of the Marginal Distributions. To answer the question of (a) how much information is encoded in the type or property set of a resource, we need to look at the marginal distributions. These provide us with the probability of a certain resource to show a particular set of types or properties. The entropy of the marginal distributions of T and R is defined as:

$$H(T) = -\sum_{t \in TS} P(T=t) \cdot \log_2 \left(P(T=t)\right) \tag{3}$$

$$H(R) = -\sum_{r \in PS} P(R=r) \cdot \log_2 \left(P(R=r)\right) \tag{4}$$

The values $H(T)$ and $H(R)$ give us an idea of how much information is encoded in the sets of types or properties of the resources. A higher value corresponds to more information, which in turn means that the sets of types or sets of properties appear more equally distributed. To be more concrete: an entropy value of 0 indicates that there is no information contained. For instance, a value of $H(T) = 0$ would indicate that all resources have exactly the same set of types (likewise for $H(R) = 0$). A maximal value, instead, is reached when the distribution is an equal distribution, i.e., each set of types or properties is equally probable. This fact also allows for normalizing the entropy values by:

$$H_0(T) = \frac{H(T)}{H^T_{\max}} = \frac{H(T)}{\log_2(|T|)} \quad , \quad H_0(R) = \frac{H(R)}{H^R_{\max}} = \frac{H(R)}{\log_2(|R|)} \tag{5}$$

The normalized entropy value ranges between 0 and 1 and indicates whether the distribution is closer to a degenerated or a uniform distribution.

Conditional Entropy. The question (b), how much information is still contained in the properties, once we know the types of a resource implies a conditional probability and, thus, a conditional entropy. We have to take a look at the distribution of the property sets given that we already know the types of a resource. The entropy in this case (i.e., the conditional entropy) conveys how much information is still in the additional observation of the properties. Again, if the set of types perfectly defines the set of properties to expect, there would be no more information to be gained. Thus, the conditional entropy would be zero. If, instead, the types were virtually independent from the properties, we would expect to observe the marginal distribution of the properties and its according entropy. Formally the conditional entropy for a given type set t is defined as:

$$H(R|T=t) = -\sum_{r\in PS} P(R=r|T=t)\log_2\left(P(R=r|T=t)\right) \tag{6}$$

$$= -\sum_{r\in PS} \frac{p(t,r)}{P(T=t)}\log_2\left(\frac{p(t,r)}{P(T=t)}\right) \tag{7}$$

Equivalently, to answer question (c), the conditional entropy for a given property set r is:

$$H(T|R=r) = -\sum_{t\in TS} \frac{p(t,r)}{P(R=r)}\log_2\left(\frac{p(t,r)}{P(R=r)}\right) \tag{8}$$

These conditional entropies are fixed to one particular set of types t or set of properties r. As we are interested in a global insight of a large scale data set like the LOD cloud, it is not feasible to look at all the individual observations. Rather we need an aggregated value.

One value of particular interest is a conditional entropy of 0. For instance, in the case of $H(R|T=t)=0$ knowing the set of types t is already conveying all the information, i.e. the set of properties can be derived with probability 1. Equivalently in the case of $H(T|R=r)=0$ we can derive the set of types from the set of properties. Accordingly we are interested in the probability of such a conditional entropy of 0, e.g. $P(H(R|T=t)=0)$ for the case of given type sets. Treating the conditional entropy itself as a random variable allows for easily estimating this probability by $P(H(R|T=t)=0)=\sum_{H(R|T=t)=0}P(T=t)$.

Expected Conditional Entropy. A similar approach is taken for the expected conditional entropy $H(R|T)$. This aggregated value also considers the conditional entropy as a random variable and computes the expected values of this variable based on the probability to actually observe a certain set of types t. The definition of this aggregation is:

$$H(R|T) = \sum_{t \in TS} P(T = t) H(R|T = t) \tag{9}$$

$$= - \sum_{t \in TS} \sum_{r \in PS} p(t, r) \log_2 \left(\frac{p(t, r)}{P(T = t)} \right) \tag{10}$$

and equivalently $H(T|R)$ is for a given set of properties r:

$$H(T|R) = - \sum_{r \in PS} \sum_{t \in TS} p(t, r) \log_2 \left(\frac{p(t, r)}{P(R = r)} \right) \tag{11}$$

Joint Entropy. Finally, we will also take a look at the joint entropy of T and R, which is defined as:

$$H(T, R) = - \sum_{t \in TS} \sum_{r \in PS} p(t, r) \log_2 (p(t, r)) \tag{12}$$

Mutual Information. To finally answer the question of (d) how far one of the schema information (either properties or types) can explain the respective other, we employ mutual information (MI) [3]. MI is a metric to capture the joint information conveyed by two random variables – and thereby their redundancy. The MI of explicit and implicit schema information of the LOD cloud is defined as:

$$I(T, R) = \sum_{r \in PS} \sum_{t \in TS} p(t, r) \log_2 \frac{p(t, r)}{P(T = t) \cdot P(R = r)} \tag{13}$$

The $\log$ expression in this sum, i.e., the expression $\log_2 \frac{p(t,r)}{P(T=t) \cdot P(R=r)}$ is also known as *pointwise mutual information* (PMI). PMI can be explained as the strength of the correlation of two events, in our case how strongly a particular type set and a particular property set are associated with each other.

One characteristics of MI is the open range of its values. A normalization of MI to the interval $[-1, 1]$ is given in [18] and involves the entropy of the marginal distributions of T and R. It is used as a direct measure for redundancy and is defined as:

$$I_0(T, R) = \frac{I(T, R)}{\min(H(T), H(R))} \tag{14}$$

4 Empirical Analysis of Linked Open Data

In the previous section, we have elaborated the metrics to obtain the relevant insights into the information and redundancy encoded in a LOD schema. In this section, we provide an approach to estimate the required probabilities from a SchemEX index structure, apply this approach to real world data and compute the metrics for our analyses.

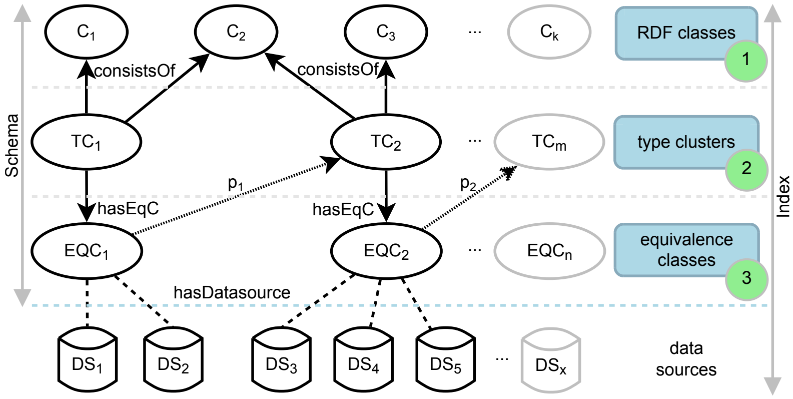

Figure 1. SchemEX index structure with three layers leveraging RDF typings and property sets

4.1 The SchemEX Index as Basis for the Analysis

The purpose of SchemEX [9,10,5] is to link schema information to data sources which provide resources conforming to this schema element. Data sources are, e.g., static RDF documents and SPARQL endpoints [7]. The central schema elements of SchemEX are Typeclusters (TC) and Equivalence classes (EQC). A TC contains all data sources which provide resources conforming to a well defined set of types/classes. The EQC divide the data sources in each TC into disjoint subsets, defined by the set of properties the instances have and in which TC the object of the triple lies. An overview of the information contained in a SchemEX index is shown in Figure 1.

It is important to note that data sources can occur in several TC or EQC as they typically describe more than one and—in particular—different kinds of resources. However, different occurrences of a data source conform to different (in particular disjoint) sets of resources. Different data volume can be reflected by annotating the data sources attached to schema elements with the *number* of resources which exhibited the according schema information [5].

Noteworthy about SchemEX is, that it can be computed very efficiently and for large data sets using a stream-based approach. In this case, the analytical component is operating in a single pass fashion over a set of RDF triples. By using a windowing technique, it is possible to obtain a very accurate schema of the processed data using commodity hardware. However, the windowing technique entails a certain loss of schema information. The extent of this loss has been analysed in detail in [4]. The type of schema information and the metrics we use in the context of this paper are relatively stable. Deviations typically range up to 5%, in single cases differences of up to 10% have been observed in an empirical evaluation.

4.2 Estimating Probabilities from a SchemEX Index

The TC elements in SchemEX [9] described in Section 4.1 correspond directly to the notion of types sets in TS given in Section 3.1. The equivalence classes in SchemEX subdivide the typeclusters and are defined by the set of properties the triples have as well as the typecluster the object of triple lies in. Hence, they are more finegrained than the property sets we are interested in. However, aggregating the equivalence classes defined by the same set of properties over all attached typeclusters, we obtain exactly the property sets PS introduced in Section 3.1. In this way we can easily construct the set PS from a SchemEX index.

As stated above, each entry in the SchemEX index refers to a distinct set of resources. Even if some of the resources are actually located in the same data source. This is provided by the pairwise disjoint character of equivalence classes. In conclusion, we can treat each entry in the index as a different set of resources, even if it is actually reflected by the same URL denoting a common data source.

If we denote with $\text{DS}(t, r)$ the set of data source entries in the SchemEX index that correspond to the resources with types t and properties r, we can estimate the above probability of observing a resource to have a particular type and property set by:

$$\hat{p}(t, r) = \frac{\sum_{d \in \text{DS}(t,r)} |d|}{N}$$

Where N is the number of all resources used to build the SchemEX and $|d|$ is the number of resources in data source d with the type set t and the property set r.

The estimates for the probabilities $p(t, r)$ above are central to all relevant metrics and effectively need only to be aggregated and normalized accordingly. However, the number of observed type sets and property sets indicates the high number of possible combinations (i.e., $|TS| \times |PS|$). The pragmatic solution to this quadratic development of combinations is not to compute all of the probabilities, but only those which actually have a non zero value. This does not affect the results of the computed metrics, as zero probabilities do not affect their overall values.

4.3 Data Sets

For our empirical analysis, we use the different segments of the data set provided for the Billion Triple Challenge (BTC) 2012. The BTC data set has been crawled from the web in a typical web spider fashion and contains about 1.44 billion triples. It is divided into five segments according to the set of URLs used as seed for the crawling process: Datahub, DBPedia, Freebase, Rest and Timbl. Details about the different parts and the crawling strategies used for collecting the data are described on the BTC 2012 data set's website[6]. As the efficient stream-based computation of a schema entails a certain loss of accuracy regarding the schema, we have to check that these inaccuracies do not affect the overall results. To this end, we use smaller data sets to compute the schema once with our stream-based approach and once in lossless approach and compare the metrics

[6] BTC 2012 data set: `http://km.aifb.kit.edu/projects/btc-2012/` (accessed 9 March 2013)

on these two schemas. As the computation of a gold standard schema has high requirements regarding the hardware resources, we were limited to derive lossless schema for data sets of up to 20 million triples. As small data sets, we used (A) the full *Rest* subset (22,328,242 triples), (B) an extract of the *Datahub* subset (20,505,209 triples) and (C) an extract of the *Timbl* subset (9,897,795 triples)[7].

The stream-based approach is also applicable to the full data crawls of (D) *Datahub*, (E) *DBPedia*, (F) *Freebase* and (G) *Timbl*. We used the same settings as in [9], using a window size of 50,000 instances for schema extraction. While the small data sets serve the purpose of confirming the stability of the stream-based approach, the larger data sets are used for the actual analysis of explicit and implicit schema information on the LOD cloud. We consider the data sets particularly useful as they span different aspects of the LOD cloud. With *Datahub*, we have got a sample of several publicly available linked RDF data sources registered in a central location. *DBpedia* is interesting as it is one of the central and most connected resources in the LOD cloud extracted from the collaboratively curated Wikipedia. *Freebase*, instead, is also a collaborative knowledge base, but here the users directly operate on the structural data. The *Timbl* data set is a crawl starting at the FOAF profile of Tim Berners-Lee (thus, the name). Hence, it provides a snapshot from yet a different part of the LOD cloud, namely starting at small, manually maintained RDF files.

5 Results of our Analysis

Table 1 gives an overview of the count statistics and metric values obtained for the smaller data sets (A), (B) and (C). The table compares the values of the lossless gold standard schema computation with the efficient stream based approach. The observed deviations in the number of type sets in the data sets (A), (B) and (C) are very low and confirm the accuracy observed in previous experiments [4]. While for the data sets (B) and (C) also the number of property sets obtained by the stream-based approach does not differ much from the gold standard, we observed a slightly stronger deviation on the Rest (A) data set. The sheer count of type and property sets, however, does not reflect the number of data sources and resources behind the individual elements in the schema. Thus, it is necessary to consider the distributions and the metrics derived from those. Here, we observe a generally quite good behaviour of the efficient schema approximation using the stream-based approach. The differences in the metrics are relatively small and consistent within each data set. In conclusion, we decided that the loss of accuracy due to the efficient stream-based schema computation is counterbalanced by the capabilities to analyse data sets which are an order of magnitude larger: the observation of more data allows for a more sound evaluation of schema information on the LOD cloud.

Table 2 gives an overview of the computed metrics on the large data sets. Already the differences in the number of observed type and property sets underline the heterogeneity of the data sets. We will now go into the details of the single metrics.

[7] The extracts correspond to the data sets that would have been obtained by stopping the crawling process after 2 hops from the Datahub URI seed set and 4 hops from the Timbl URI seed set. We did not produce extracts for DBpedia and Freebase as the hop information is not provided for these BTC subsets.

Table 1. Statistics of the schema information obtained for the smaller data sets when using lossless and efficient (stream-based) schema computation.

Data set		**(A) Rest**		**(B) Datahub (extract)**		**(C) Timbl (extract)**	
Number of Triples		22.3M		20.5M		9.9M	
Schema construction		lossless	efficient	lossless	efficient	lossless	efficient
Type sets	$\|T\|$	791	793	3,601	3,656	1,306	1,302
Property sets	$\|R\|$	8,705	7,522	4,100	4,276	3,015	3,085
Entropy of type sets	$H(T)$	2.572	2.428	3.524	3.487	2.839	2.337
Normalized entropy of type sets	$H_0(T)$	0.267	0.252	0.298	0.295	0.274	0.226
Entropy of property sets	$H(R)$	4.106	4.708	6.008	6.048	3.891	3.258
Normalized entropy of property sets	$H_0(R)$	0.314	0.366	0.501	0.501	0.337	0.281
Expected conditional entropy, given properties	$H(T\|R)$	0.295	0.289	1.158	1.131	0.670	0.512
Probability of $H(T\|R = r) = 0$	$P(H(T\|R = r) = 0)$	29.32%	38.02%	60.77%	57.79%	27.81%	21.52%
Expected conditional entropy, given types	$H(R\|T)$	1.829	2.568	3.643	3.692	1.723	1.433
Probability of $H(R\|T = t) = 0$	$P(H(R\|T = t) = 0)$	6.22%	5.31%	12.01%	11.08%	6.06%	4.51%
Joint entropy	$H(T, R)$	4.401	4.997	7.166	7.179	4.561	3.770
Mutual Information	$I(T, R)$	2.277	2.140	2.365	2.356	2.169	1.824
Normalized Mutual Information	$I_0(T, R)$	0.885	0.881	0.671	0.676	0.764	0.781

Table 2. Statistics of the schema information obtained for the full data sets when using efficient (stream-based) schema computation.

Data set		**(A)** **Rest**	**(D)** **Datahub (full)**	**(E)** **DBpedia**	**(F)** **Freebase**	**(G)** **Timbl (full)**
Number of Triples		22.3M	910.1M	198.1M	101.2M	204.8M
Type sets	$\lvert T\rvert$	793	28,924	1,026,272	69,732	4,139
Property sets	$\lvert R\rvert$	7,522	14,712	391,170	162,023	9,619
Entropy of type sets	$H(T)$	2.428	3.904	1.856	2.037	2.568
Normalized marginal entropy of type sets	$H_0(T)$	0.252	0.263	0.093	0.127	0.214
Entropy of property sets	$H(R)$	4.708	3.460	6.027	2.868	3.646
Normalized entropy of property sets	$H_0(R)$	0.366	0.250	0.324	0.166	0.276
Expected conditional entropy, given properties	$H(T\lvert R)$	0.289	1.319	0.688	0.286	0.386
Probability of $H(T\lvert R = r) = 0$	$P(H(T\lvert R = r) = 0)$	38.02%	11.59%	54.85%	80.89%	15.15%
Expected conditional entropy, given types	$H(R\lvert T)$	2.568	0.876	4.856	1.117	1.464
Probability of $H(R\lvert T = t) = 0$	$P(H(R\lvert T = t) = 0)$	5.31%	10.83%	3.73%	2.05%	1.60%
Joint entropy	$H(T, R)$	4.997	4.779	6.723	3.154	4.032
Mutual Information	$I(T, R)$	2.140	2.585	1.178	1.751	2.182
Normalized Mutual Information	$I_0(T, R)$	0.881	0.747	0.635	0.860	0.850

Entropy in Type and Property Sets. We can observe the tendency that the property sets convey more information than type sets. This can be observed in the higher values of the normalized entropies. For instance, the normalized marginal entropy of the property sets has a value of 0.324 on the *DBpedia* (E) data set, while the normalized marginal entropy of the type sets is 0.093. This observation provides a hint that on *DBpedia* the distribution into type sets is far more skewed than the distribution of property sets. Similar observations can be made for the data set (A), (F) and (G), though to a lower extent. An exception is the *Datahub* data set (D), where the distribution of resources in type sets and property sets seems comparable.

Conditional Entropies. Looking at the expected conditional entropies reveals some interesting insights. Recall that the aggregation we chose for the conditional entropy provides us with the expected entropy, given a certain type set or property set. We can see in Table 2 that the entropy given a property set tends to be far lower than the one when given a type set. In conclusion: knowing the properties of a resource in these cases already tells us a lot about the resource, as the entropy of the conditional distribution can be expected to be quite low. On the contrary, when knowing the type of a resource the entropy of the distribution of the property sets can be expected to be still relatively high (when compared to the entropy of the marginal distribution). We looked at the data more closely to investigate how often a given type set is already a clear indicator for the set of properties (and vice versa). This insight is provided by considering the probabilities $P(H(R|T = t) = 0)$ and $P(H(T|R = r) = 0)$ to observe a conditional entropy of 0. The most extreme case is the *Freebase* (F) data set, where for 80.89% of all resources it is sufficient to know the set of properties in order to conclude the set of types associated with this resource. Knowing, instead, the types of a resource conveys less information: only in 2.05% of the cases this is sufficient to predict the set of properties of a resource. Again, and with the exception of *Datahub* (D), the other data sets exhibit a similar trend. However, at very different levels: the probability of knowing the type set for a given property set ranges between 15.15% and 54.85%. The *Datahub* data set shows a far more balanced behaviour. Both probabilities $P(H(R|T = t) = 0)$ and $P(H(T|R = r) = 0)$ are at around 11%, confirming the particular form of this data set.

Mutual Information. Finally, the value of the normalized MI gives us insights on how much one information (either properties or types) explains the respective other. Also here, we observe a quite wide range from 0.635 on *DBpedia* (E) to 0.881 on *Rest* (A). Accordingly, extracting only type or only property information from LOD can already explain a quite large share of the contained information. However, given our observations a significant part of the schema information is encoded also in the respective other part. The degree of this additional information depends on the part of the LOD cloud considered. As a rule of thumb, we hypothesise that collaborative approaches without a guideline for a schema (such as *DBpedia*) tend to be less redundant than data with a narrow domain (*Timbl*) or some weak schema structure (*Freebase*).

Discussion of the Results. The observations on the large data sets provide us with insights into the form and structure of schema information on the LOD cloud. First of all, the distribution of type sets and property sets tend to have a relatively high normalized entropy. We can conclude that the structure of the data is not dominated by a few combinations of types or properties. Accordingly for the extraction of schema information, we cannot reduce the schema to a small and fixed structure but need to consider the wide variety of type and property information. Otherwise the schema would loose too much information.

A second observation is the dependency between types and properties. The conditional entropy reveals that the properties of a resource usually tell much more about its type than the other way around. This observation is interesting for various applications. For instance, suggesting a data engineer the types of a resource based on the already modelled properties seems quite promising. We assume that this observation can also be seen as an evidence that property information on the LOD cloud actually considers implicit or explicit agreements about the domain and range of the according property. However, this observation is not valid for the entire LOD cloud. Depending on the concrete setting and use case, a specific analysis might need to be run.

Finally, the observed MI values underline the variance of schema information in the LOD cloud. Ranges from 63.5% to 88.1% redundancy between the type sets and property sets have been observed. Thus, approaches building a schema only over one of these two types of schema information run at the risk of a significant loss of information.

6 Conclusions and Future Work

In this paper, we have proposed a method and metrics for conducting in depth analysis of schema information on Linked Open Data. In particular, we have addressed the question of dependencies between the types of resources and their properties. Based on the five segments of the BTC 2012 data set, we have computed various entropy metrics as well as mutual information. In conclusion, we observe a trend of a reasonably high redundancy between the types and properties attached to resources. As more detailed conclusion, we can derive that the properties of a resource are rather indicative for the type of the resource. In the other direction, the indication is less strong. However, this observation is nor valid for all sources on the LOD cloud. In conclusion, if the application and data domain is not known, it is necessary to capture both: explicit and implicit schema information.

As future work, we plan to deepen these insights and incorporate the obtained deeper understanding into various applications. Therefore, we will look into the details of the conditional distributions for given type sets and property sets. In this way, we might identify which sets of types and properties allow for highly precise predictions of the respective other schema information. On the application side, we plan to use the gained insights for various purposes: index compression for SchemEX as well as the detection of schema patterns that are stable enough—and thereby suitable—for constructing an API for accessing LOD resources.

Acknowledgements. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257859, ROBUST.

References

1. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing linked datasets with the void vocabulary. `http://www.w3.org/TR/void/`, (accessed 9 March 2013)
2. Auer, S., Demter, J., Martin, M., Lehmann, J.: Lodstats – an extensible framework for high-performance dataset analytics. In: Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) Knowledge Engineering and Knowledge Management, Lecture Notes in Computer Science, vol. 7603, pp. 353–362. Springer Berlin Heidelberg (2012)
3. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience (1991)
4. Gottron, T., Pickhardt, R.: A detailed analysis of the quality of stream-based schema construction on linked open data. In: CSWS'12: Proceedings of the Chinese Semantic Web Symposium (2012), to appear
5. Gottron, T., Scherp, A., Krayer, B., Peters, A.: Get the google feeling: Supporting users in finding – relevant sources of linked open data at web-scale. In: Semantic Web Challenge, Submission to the Billion Triple Track (2012)
6. Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K.U., Umbrich, J.: Data summaries for on-demand queries over linked data. In: WWW. pp. 411–420. ACM (2010)
7. Heath, T., Bizer, C.: Linked Data: Evolving the Web Into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool (2011)
8. Hogan, A., Umbrich, J., Harth, A., Cyganiak, R., Polleres, A., Decker, S.: An empirical survey of linked data conformance. Web Semantics: Science, Services and Agents on the World Wide Web 14(0), 14 – 44 (2012)
9. Konrath, M., Gottron, T., Scherp, A.: Schemex – web-scale indexed schema extraction of linked open data. In: Semantic Web Challenge, Submission to the Billion Triple Track (2011)
10. Konrath, M., Gottron, T., Staab, S., Scherp, A.: Schemex—efficient construction of a data catalogue by stream-based indexing of linked data. Web Semantics: Science, Services and Agents on the World Wide Web 16(0), 52 – 58 (2012), the Semantic Web Challenge 2011
11. Lorey, J., Abedjan, Z., Naumann, F., Böhm, C.: Rdf ontology (re-) engineering through large-scale data mining. In: Semantic Web Challenge (2011)
12. Luo, X., Shinavier, J.: Entropy-based metrics for evaluating schema reuse. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) The Semantic Web, Lecture Notes in Computer Science, vol. 5926, pp. 321–331. Springer Berlin Heidelberg (2009)
13. Maduko, A., Anyanwu, K., Sheth, A., Schliekelman, P.: Graph summaries for subgraph frequency estimation. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) The Semantic Web: Research and Applications, Lecture Notes in Computer Science, vol. 5021, pp. 508–523. Springer Berlin Heidelberg (2008)
14. Neumann, T., Moerkotte, G.: Characteristic sets: Accurate cardinality estimation for rdf queries with multiple joins. In: Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany. pp. 984–994 (2011)
15. Neumann, T., Weikum, G.: Scalable join processing on very large rdf graphs. In: SIGMOD Conference. pp. 627–640. ACM (2009)
16. Schaible, J., Gottron, T., Scheglmann, S., Scherp, A.: LOVER: Support for Modeling Data Using Linked Open Vocabularies. In: LWDM'13: 3rd International Workshop on Linked Web Data Management (2013), to appear
17. Shannon, C.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423 and 623–656 (July and October 1948)
18. Yao, Y.: Information-theoretic measures for knowledge discovery and data mining. In: Karmeshu (ed.) Entropy Measures, Maximum Entropy Principle and Emerging Applications, Studies in Fuzziness and Soft Computing, vol. 119, pp. 115–136. Springer Berlin Heidelberg (2003)