# Profiling Linked Open Data with ProLOD

Christoph Böhm[1], Felix Naumann[1], Ziawasch Abedjan[2], Dandy Fenz[2]
Toni Grütze[2], Daniel Hefenbrock[2], Matthias Pohl[2], David Sonnabend[2]

*Hasso-Plattner-Institut; Prof.-Dr.-Helmert-Str. 2-3; D-14482 Potsdam, Germany*
[1]firstname.lastname@hpi.uni-potsdam.de
[2]firstname.lastname@student.hpi.uni-potsdam.de

***Abstract*— Linked open data (LOD), as provided by a quickly growing number of sources constitutes a wealth of easily accessible information. However, this data is not easy to understand. It is usually provided as a set of (RDF) triples, often enough in the form of enormous files covering many domains. What is more, the data usually has a loose structure when it is derived from end-user generated sources, such as Wikipedia. Finally, the quality of the actual data is also worrisome, because it may be incomplete, poorly formatted, inconsistent, etc.**

**To understand and profile such linked open data, traditional data profiling methods do not suffice. With ProLOD, we propose a suite of methods ranging from the domain level (clustering, labeling), via the schema level (matching, disambiguation), to the data level (data type detection, pattern detection, value distribution). Packaged into an interactive, web-based tool, they allow iterative exploration and discovery of new LOD sources. Thus, users can quickly gauge the relevance of the source for the problem at hand (e.g., some integration task), focus on and explore the relevant subset.**

## I. Profiling Linked Open Data

Data profiling comprises a well established set of basic operations, which analyze a (relational) dataset and create metadata that is useful to understand the data and to detect irregularities. Profiling is mostly performed in a column-by-column manner, for instance to detect frequent value patterns or the uniqueness of column values. Common profiling methods and tools have the underlying assumption of a well-defined semantics of the column and mostly regular data.

These assumptions do not hold for linked open data (LOD) published on the web. Such data emerge from different sources, such as open source communities (e.g., Wikipedia) or projects dedicated to a specific topic (e.g., DrugBank [1]). These diverse origins cause a diversity of how information is expressed as data values and how these values are structured. Nevertheless, these datasets interlink each other. The overall LOD vision is to enable the generation of new knowledge based on a wealth of widely available interlinked data. However, leveraging the variety of such open data requires (i) an initial understanding of each single dataset and (ii) an overview of the available data as a whole. Only then, data analysts can focus on the required subset of LOD for the problem at hand. Classical profiling techniques are, to the best of our knowledge, not appropriate to deal with these new massive sets of open (and thus heterogeneous) data. We propose a new iterative and interactive methodology for profiling LOD. We envision a process that allows a user to divide data into groups, review simple statistics or sophisticated mining results on a group-level, and then rethink grouping decisions in order to revise them for refining the profiling result. In this paper, we report on ProLOD, an initial prototype we developed to step towards this vision. As a proof-of-concept, we concentrate on the infobox (without ontology mappings) and short abstract data subset of DBpedia 3.2 [2]. DBpedia is generated from Wikipedia and comprises 34.2 million triples.

## II. Related Work

Technologies for dealing with large unstructured datasets are currently under development in two distinct communities, namely the Semantic Web community as well as the database community where relevant techniques emerge from the *Dataspace* area. Such large datasets are, on the one hand, published on the Web according to a set of best practices [3], whereas, on the other hand, the data stems from heterogenous sources within a specific scope, such as enterprises, or scientific collaborations [4].

Currently, several projects attempt to visualize heterogeneous data. We briefly discuss two examples: *Tabulator* [5] is an RDF browser that is designed to provoke interest in the Semantic Web for new users as well as to support instant feedback for developers. The authors promote two different views – *Explore* and *Analyze*. The first allows the traversal of an RDF graph while the latter enables the analysis of aggregated data that match a query. The *Vispedia* project [6] is an approach to interactively visualize Wikipedia infoboxes. It allows a user to select an infobox and define a keyword query, which the system evaluates on the semantic graph of Wikipedia to extract supplemental information. The result is presented on a map, a timeline, or in a graph.

Some database research also deals with huge amounts of triples since triples are a generic way for representing data. Here, the literature deals with indexing [7], or techniques for property and path analysis with limited schema information [8]. It also discusses storage systems [9] as well as query-, and data models [8], [10].

As for data profiling in general, there is a wealth of approaches that can be used to grasp a given dataset, e.g., functional dependency discovery [6], and join path exploration [11]. The *Bellman* project [12] integrates a set of techniques to address poorly structured and dirty data. Additionally, there are numerous tools from commercial vendors, [13], [14]. However, these tools are inappropriate for unstructured and heterogeneous datasets, such as DBpedia: (1) Analysis focuses on (one or more) columns, which are not present here. (2) Mining a dataset's structure is not possible. (3) Analyses assume some sort of clean data, which a data analyst aims to understand – not a dataset that needs to be structured, cleansed as well as understood at the same time.

## III. Clustering and Labeling

Clustering is a preliminary requirement when dealing with large and heterogeneous datasets, because millions of triples as a whole do not allow to derive meaningful results. A predicate, such as length, could appear in a large variety of contexts and domains. Calculating value distributions over all its values can be meaningless. We argue that a good clustering can divide the data into semantically correlated data(sub)sets, for which meaningful and insightful profiling results are possible. As a side-effect, clusters partition data into smaller subsets, better enabling on-the-fly profiling. Difficulties include the unknown number of clusters, the size of the dataset and the meaningful labeling of discovered clusters.

Our prototype provides an initial pre-calculated hierarchical clustering of the dataset to profile, which can be refined interactively at runtime. The root cluster represents the complete dataset and each cluster can itself be divided into several subclusters. In our prototype, the clustering component is the only part that performs computation on the entire dataset – namely while computing the initial clusters. All further profiling steps are implicitly performed only on entities of the cluster in focus.

In order to deal with the potentially enormous number of triples during clustering, we employ several strategies. (1) The initial clustering is computed in a pre-processing step. (2) Clustering is done using a schema-based similarity measure. Thus, it only needs to take into account the existence of predicates of an entity, not their actual values. (3) During clustering, our prototype employs several heuristics; for example, it uses a cluster-internal dissimilarity threshold to decide whether to further split a cluster into subclusters. Clustering is based on the well known *K-Means* algorithm, which is fast and has been shown to yield near-to-optimum results, in most cases. Since the number of clusters is unknown, we have implemented a version that iteratively increments the number of clusters, as well as one version that recursively invokes itself on clusters in order to create a cluster hierarchy.

For each cluster, we compute a *mean schema*: It consists of a set of predicates that together form a schema that is distinctive for its cluster and, therefore, constitutes a brief summary for the structure of its entities. Currently, the mean schema is computed during clustering, where the $n$ most frequently occurring predicates of a cluster are selected for its mean schema. We determine $n$, the size of the mean schema, as the average schema size of all entities in that cluster.

To give users more insightful feedback about the content of the clusters, a labeling is required. Because all entities of a cluster are semantically correlated, they should have some common characteristics. In general, many entities have some textual descriptions as rdfs:comment or some equivalent properties. Since we do not want to concentrate on a predefined setting, we treat all strings of length $l$ as texts describing an entity. Here, manual reviews have shown that we get good results for $l = 100$. We compute *tf-idf* values to determine the *m* most important terms of each cluster and use them as cluster label. Table I shows some examples. Alternatively, if no textual description could be found, the Top-*m* properties of the cluster mean-schema are used as label.

| Label | Sample subjects |
|---|---|
| *minister politician mayor* | Angela_Merkel<br>Ted_Kennedy<br>Presidency_of_George_Washington |
| *film directed starring* | Titanic_(1997_film)<br>Metropolis_(film)<br>Frankenstein_(1910_film) |
| *club football league* | FC_Bayern_Munich<br>Liverpool_F.C.<br>Los_Angeles_Galaxy |

TABLE I

Exemplary cluster labels and corresponding subjects for *m*=3

## IV. Schema Discovery

After having clustered the data into semantically correlated subsets, a natural next step is to perform more detailed schema discovery for each of these clusters beyond the mean schema. Such schema discovery enables initial understanding of the actual structure of the data, because a set of triples does not expose much structural information.

We propose a process that includes determining the actual schema (e.g., distinct attributes of a cluster), finding equivalent attributes (e.g., name, family name, and surname), and discovering poor attributes, i.e., those that do not contain useful values for most data entries. More sophisticated analyses discover attribute correlations, such as association rules, inverse relations, or foreign key relationships. Note that *attributes* mean predicates that have a clean and unified semantics.

Our tool detects positive and negative association rules that show occurrence dependencies among predicates within a cluster. As for a *book* entity from a *media* cluster, such combinations with high combined occurrence are author, isbn, and genre. To detect these predicate cooccurrences, a version of the Apriori Algorithm [15] is used. Afterwards, a straightforward algorithm generates all rules that hold a desired confidence and correlation coefficient $\rho$. Table II shows a list of generated rules that hold in the *media* cluster for the aforementioned examples.

| Rule | Confidence | Correlation Coefficient |
|---|---|---|
| $genre, isbn \Rightarrow author$ | 0.99 | 0.67 |
| $isbn \Rightarrow author$ | 0.92 | 0.66 |
| $isbn \Rightarrow author, genre$ | 0.83 | 0.66 |
| $author, genre \Rightarrow isbn$ | 0.70 | 0.66 |
| $author \Rightarrow isbn$ | 0.64 | 0.66 |
| $author \Rightarrow genre, isbn$ | 0.58 | 0.67 |

TABLE II

Example rules with 10% support, 50% confidence, and $\rho = 0.4$

A second way of using association rules is to discover negative dependencies among predicates [16]. Such negative rules hint at heterogeneity among the schemata of entities in the current cluster. Either the cluster is poorly built and contains entities from different domains, or the negatively associated predicates have equivalent semantic meanings and therefore never occur together. For instance, we discovered that the predicates author and developer are negatively associated in the *media* cluster. The rules $author \Rightarrow \neg developer$ and $developer \Rightarrow \neg author$ have a confidence above 90%. In the spirit of holistic matching [17], we could conclude semantic equivalence. In fact, ProLOD allows to mark two attributes within a cluster as equivalent, and further mining will always combine the values from both. The negative association is also true for the predicates title and name, which may lead to the assumption that the *media* cluster contains different types of media, such as novels and computer games. Hence, using standard procedures to find positive and negative association rules, it is possible to combine results of both methods and scrutinize assumptions about the schemata of entities. Furthermore, finding association rules within the entire, unclustered dataset may lead to new ideas for clusters. For example, highly negative associated schemata would be useful hints for selecting seeds for the K-Means algorithm (which is left for future work).

Another proposal to extract schema information is to examine linked entities. If subject $X$ holds a link to subject $Y$ via predicate $A$ ($X \xrightarrow{A} Y$), one can denote a semantic connection. A stronger form of interconnection between two potential correlated entities are links with inverse predicates $A$ and $B$. So if two links $X \xrightarrow{A} Y$ and $Y \xrightarrow{B} X$ exist, one can state that these two entities are considerably connected. The more entities are connected via predicates $A$ and $B$, the higher the correlation of the inverse predicate pair. In DBpedia 3.2, there are approx. 13m different links between entities; 4% of them form inverse predicate pairs. An example is the relation among the book entity *Into the Wild* and the author entity *Jon Krakauer*: Into the Wild $\xrightarrow{debutWorks}$ Jon Krakauer and Jon Krakauer $\xrightarrow{author}$ Into the Wild. Table III shows further examples. Notice that [before,after] is the most frequent inverse link pair, whereas its correlation coefficient is not as high as others, because of the remaining frequent independent use of both predicates. Additionally, there are some occurrences of before with itself, but the correlation coefficient is negative. This may point to inconsistencies in the data.

| $\overrightarrow{PredicateA}$ | $\overleftarrow{PredicateB}$ | Corr Coef | Frequency |
|---|---|---|---|
| before | after | 0.239 | 28856 |
| sisterStations | sisterStations | 0.749 | 7494 |
| precededBy | followedBy | 0.830 | 7097 |
| spouse | spouse | 0.322 | 1964 |
| before | before | -0.003 | 738 |
| star | exoplanet | 0.895 | 188 |

TABLE III
EXAMPLE INVERSE PREDICATES (ORDERED BY FREQUENCY)

Like in the debutWorks example mentioned before, many links connect entities of different types and, accordingly, schemata as well as clusters. In conclusion, analyses of links and inverse predicates can lead to statistics about inter- and intra-cluster relationships, which help to perform a semantic (pre-)clustering. Additionally, the existence of inverse predicates may indicate the existence of redundant information about the same fact (e.g., followedBy and precededBy, or spouse with itself).

## V. Data Types and Pattern Statistics

Having a set of clusters and some knowledge about its structure, traditional profiling methods are applicable. Here, we gather statistics about data types and pattern distributions by analyzing the object values of the RDF triples. In Wikipedia, many users modify articles and use different syntax to represent the same fact. In addition, predicates are often misused when no applicable predicate exists. Profiling statistics support the detection of such discordant values or misused predicates and facilitate to find valid formats for specific predicates.

A characteristic of linked data is the fact that an object value can be an *internal link* to another subject, an *external link* or a *literal*. Internal links are pointers to data within the local source; we consider only literal values and external links for further statistics, since we find that computing basic values for internal links is not useful. Profiling internal links is left for future work. At this point, we do also not yet follow external links for simplicity reason. Literals can be of different data types. Here, type detection using regular expressions is the first layer of profiling object values.

In the next step, we determine the value distribution within these data types. Values of numeric data types are divided into ranges. For non-numeric data types, we generate patterns and determine their frequencies. To visualize huge numbers of different patterns (even for a single predicate), we introduce *normalized patterns*. Such normalized patterns summarize patterns by identifying sequences of the same character and reducing these sequences to a single character. Furthermore, ProLOD is able to drill down to the actual data values represented by the pattern or numeric range.

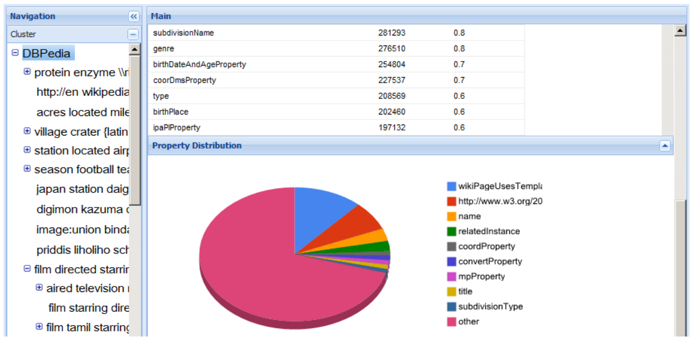
ProLOD statistics are best illustrated by the example predicate zipcode. Usually, zip codes are sequences of digits, i.e., data type Integer is expected. Due to DBPedia's heterogeneous data, String data was also discovered. One normalized pattern found is a-9, which represents all strings that have a sequence of letters followed by a sequence of digits separated by a sequence of hyphens. In our example, the only pattern normalized this way is AA-99999, which captures zip codes preceded by a country code. For this, it is useful to leverage ProLOD's drill-down functionality to find out such information.

## VI. The ProLOD Tool

ProLOD is a web-based prototype. We store the triples to profile in a relational database to be able to perform basic database operations. The profiling process is divided into two phases: pre-processing and realtime profiling.

The purpose of the pre-processing phase is to build additional data structures that enable realtime profiling in the second phase. During pre-processing, our prototype executes several computations on the input data: (1) The triples are transformed and stored in a normalized database schema. (2) Each triple is enriched with data type and pattern information. (3) An initial clustering is computed and cluster-labels are determined. Since the pre-processing phase is computationally intensive and not yet highly optimized the DBpedia dataset load and preprocessing takes about one day.

In the realtime profiling phase, the user interacts with the web interface, shown in Figure 1. The interface is divided into a cluster tree view on the left and a details view on the right. The cluster view enables users to explore the cluster tree and to select a cluster for further investigation. This view can also be used to modify the clustering by creating new clusters, merging existing ones, or reclustering with a different $k$. After selecting a cluster, the user can find facts and statistics about it in the details view, which can also be used to perform schema discovery. Further, it allows cluster modification through additional filtering.



| | | |
|---|---|---|
| subdivisionName | 281293 | 0.8 |
| genre | 276510 | 0.8 |
| birthDateAndAgeProperty | 254804 | 0.7 |
| coorDmsProperty | 227537 | 0.7 |
| type | 208569 | 0.6 |
| birthPlace | 202460 | 0.6 |
| ipaPlProperty | 197132 | 0.6 |

Fig. 1. The ProLOD web interface.

## VII. Summary and Outlook

Data profiling, when applied to heterogeneous data, must be extended beyond the typical tasks of determining value distributions and patterns, finding outliers in columns, etc. Rather, profiling tools need to enable users to create an assessment of the data at hand. This begins at source-level, where clustering can help to identify the domain(s) of the data and their representative schema. The need for assessment continues at attribute-level, where attributes can store data of wildly different semantics and values of same semantics are spread over many attributes. Finally, traditional data profiling tasks apply.

Due to the heterogeneity of the underlying data, a profiling tool can no longer have a mere reporting functionality; it is necessary to allow users to interactively learn insights about the data during subsequent profiling steps. Hence, performance can no longer concentrate on offline reporting, but must allow user feedback and perform on-the-fly profiling.

With ProLOD, we have developed a corresponding prototype, concentrating on the DBpedia infobox dataset. In its current form, it is only a starting point, implementing a subset of possible tasks[1]. An important aspect of future work is scalability to even larger datasets. While DBpedia with its 34 million triples is already formidable, the Billion Triple Challenge[2] hints at typical future data volumes. Finally, we plan to examine possible uses of ProLOD, for instance to help Wikipedia authors in writing infoboxes or to enable rapid integration of LOD sources.

**Acknowledgements.** This research was supported in part by the German Research Society (DFG grant no. NA 432). Additionally, we thank Christian Bizer and Georgi Kobilarov for providing DBpedia to the community and for their valuable feedback on ProLOD.

## References

[1] D. S. Wishart, C. Knox, A. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, and M. Hassanali, "DrugBank: a knowledgebase for drugs, drug actions and drug targets," *Nucleic Acids Research*, vol. 36, pp. 901–906, 2008.
[2] C. Bizer, J. Lehmann, S. A. Georgi Kobilarov, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia A Crystallization Point for the Web of Data," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, p. 154165, 2009.
[3] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data - The Story So Far," *Int. Journal on Semantic Web and Information Systems*, 2009.
[4] M. Franklin, A. Halevy, and D. Maier, "From Databases to Dataspaces: A New Abstraction for Information Management," *SIGMOD Record*, vol. 34, pp. 27–33, 2005.
[5] T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets, "Tabulator: Exploring and analyzing linked data on the semantic web," in *Proc. of the 3rd Int. Semantic Web User Interaction Workshop*, 2006.
[6] B. Chan, L. Wu, J. Talbot, M. Cammarano, and P. Hanrahan, "Vispedia: Interactive Visual Exploration of Wikipedia Data via Search-Based Integration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, pp. 1213–1220, 2008.
[7] X. Dong and A. Halevy, "Indexing Dataspaces," in *Proc. of the ACM Int. Conf. on Management of Data (SIGMOD)*, 2007, pp. 43–54.
[8] B. Howe, D. Maier, N. Rayner, and J. Rucker, "Quarrying dataspaces: Schemaless profiling of unfamiliar information sources," in *Proc. of the ICDE Workshop on Information Integration Methods, Architectures, and Systems (IIMAS)*, 2008, pp. 270–277.
[9] D. J. Abadi, A. Marcus, S. R. Madden, and K. Hollenbach, "Scalable semantic web data management using vertical partitioning," in *Proc. of the Int. Conf. on Very Large Databases (VLDB)*, 2007, pp. 411–422.
[10] A. Balmin and E. Curtmola, "Wikianalytics: Ad-hoc querying of highly heterogeneous structured data," in *Proc. of the Int. Conf. on Data Engineering (ICDE) Demo*, 2009.
[11] C. M. Procopiuc and D. Srivastava, "Database Exploration Using Join Paths," in *Proc. of the Int. Conf. on Data Engineering (ICDE)*, 2008, pp. 1331–1333.
[12] T. Johnson, A. Marathe, and T. Dasu, "Database Exploration and Bellman," *IEEE Data Engineering Bulletin*, vol. 26, pp. 34–39, 2003.
[13] "IBM InfoSphere Information Analyzer," http://www.ibm.com/software/data/infosphere/information-analyzer, last access: 07/2009.
[14] "ORACLE DATA PROFILING," http://www.oracle.com/technology/products/oracle-data-quality, last access: 07/2009.
[15] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," in *Proc. of the Int. Conf. on Very Large Databases (VLDB)*, 1994, pp. 487–499.
[16] M.-L. Antonie and O. R. Zaïane, "Mining positive and negative association rules: an approach for confined rules," in *European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD)*, 2004, pp. 27–38.
[17] S.-L. Chuang and K. C.-C. Chang, "Integrating web query results: holistic schema matching," in *Proc. of the Int. Conf. on Information and Knowledge Management (CIKM)*, 2008, pp. 33–42.

[1]A screencast is available at `http://tinyurl.com/prolod-01`
[2]`http://challenge.semanticweb.org`