Towards Sustainable Extract-Transform-Load (ETL) Fusion of Company Data

Kay Müller, Claus Stadler, Ritesh Kumar Singh, Sebastian Hellmann

AKSW/KILT, Leipzig University & DBpedia Association
{kay.mueller,cstadler,ritesh.kumar.singh,hellmann}@informatik.uni-leipzig.de
http://aksw.org/Groups/KILT.html

Abstract. Company Data providers range from government agencies, over commercial enterprises to communities of data enthusiasts. Hence the amount of information and the quality of these datasets varies. Integrating different company datasets into one graph by using ETL (Extract-Transform-Load) systems which perform offline transformation, ontology matching and linking techniques can result in target datasets which contain many mapping, linking and consistency errors. Since ETL systems produce the RDF offline, any mapping or content change requires a re-ingest of the relevant source data. When dealing with heterogeneous source datasets, such as the company datasets, creating a unified target dataset can be a tedious undertaking. Therefore the paper proposes an RDF view based ingestion approach, which allows real-time "debugging" of the unified dataset where mappings and links can be changed with immediate effect. Once the unified graph passes all data quality tests, the RDF can be materialized. This process poses an alternative to existing ETL solutions.

Keywords: Company Data, Linked Data, ETL, RDF View

1 Introduction and Related Work

While some countries have decided to publicly release official data about their companies, most countries –unfortunately– do not release this information freely.[1] Therefore, private organizations and data enthusiasts are collecting company data as well. Examples of the second group are OpenCorporates[2], Thomas Reuters Permid[3], Crunchbase[4], DBpedia[5] and many more. As one can imagine, the company dataset content and quality differs between these datasets. Furthermore, no standard source file format or schema has been defined, which makes it difficult to map and link the different datasets to a unified dataset.

[1] http://index.okfn.org/dataset/companies
[2] https://opencorporates.com
[3] https://permid.org
[4] https://www.crunchbase.com
[5] http://dbpedia.org

Also, information about companies can get out of date very quickly: For example, a company may be newly founded, cease to exist, appoint a new CEO, or change its name. Traditionally ETL (Extract-Transform-Load) systems are used to convert different source datasets into a structured target dataset. These systems are commonly used in the area of data warehousing. As the name suggests these frameworks follow the workflow of: 1.) *Extract* required information from source datasets, 2.) *Transform* portions of the source data into a target model using schema mappings, normalizations and deduplications, 3.) *Load* the data into a store, possibly with versioning support.

In the Semantic Web community tools such as LDIF[6], OpenRefine[7], Unified Views[8] have been developed to allow Linked Data developers to create new Linked Data content. Despite their success, some problems still exist. Often, Linked Data data publishers do not follow existing best practices and (quasi-)standards, which e.g. complicates linking tasks and overall accounts for a decreased data quality. The LOD Laundromat [1] was created due to this dilemma: This system crawls the Linked Data Cloud[9] and creates corresponding standard-compliant datasets which can be downloaded from the website. Since data quality can vary between datasets, using an ETL system to convert the source data into the target format might result in mapping and content errors, which might only be discovered once all the data is converted. Very often this might result in a re-ingest of the source data through the whole ETL pipeline. These trial and error attempts can be very time and resource consuming. As was shown in [2] RDF views can present a feasible alternative to standard ETL approaches in the context of data quality management. Based on the concept of RDF views, this paper proposes a novel architecture which gives the possibility to "debug" knowledge graphs, hence support the edit of links, mappings and data cleaning components where the effect of these changes can be observed immediately. With the help of this novel architecture real-time entity fusion and better handling of meta-information can be achieved.

2 Design

In order to ease the quality assessment of heterogeneous company dataset, we propose a view-based RDF transformation design. This design bears the advantage that many data and transformation related changes can be reviewed promptly instead of potentially having to re-ingest the data. The proposed design is shown in Figure 1. In order to support the handling of heterogeneous company datasets we identified the following main features:

- *RDF views*: Since our proposed architecture uses RDF-based views, we suggest to convert the source data directly to RDF without cleaning the input data. If data from the same data provider is re-ingested, hashing algorithms will ensure that only changed/new data is loaded into the graph. As

[6] `http://ldif.wbsg.de/`
[7] `http://openrefine.org/`
[8] `https://www.semantic-web.at/unifiedviews`
[9] `http://lod-cloud.net/`

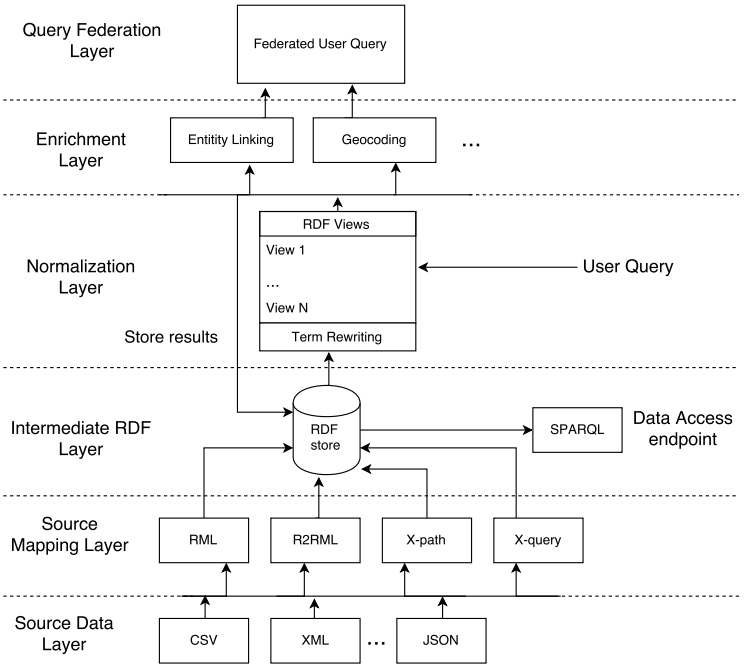

Fig. 1: RDF Views Design Diagram

RDF2RDF mapping languages, SPARQL CONSTRUCT[10] queries with minor syntax extensions for naming views and quad support could be exploited. Another option is to adapt RML[11] for this purpose.

- Functional Indexes for transformation functions: The input data is rarely expected to fit the target ontology and all defined requirements. In order to speed up access to data normalized via views, the proposed system supports functional indices similar to those supported by traditional RDBMS.[12].
- *Real-time updates*: Due to the virtualization of the RDF and Normalization Layer, changes applied in these layers have immediate effect on the virtualized RDF views. This feature allows ontology engineers to follow an iterative process when creating a new dataset, rather than re-ingesting and checking parts of the dataset in order to save time.
- *Unified Querying and ETL*: All relevant source data is expected to have been mapped to appropriate target ontologies. Using SPARQL, it is possible to retrieve portions of the data on-the-fly as well as to export all data at once.

Our design consists of 6 layers: The *Source Data Layer* is used to ingest the original source data via the *Source Mapping Layer* into the RDF backend

[10] `https://www.w3.org/TR/rdf-sparql-query/#construct`
[11] `http://semweb.mmlab.be/rml/spec.html`
[12] For example, see Postgres: `http://www.postgresql.org/docs/9.5/static/indexes-expressional.html`

using mappings which alter the source data as little as possible. The *Intermediate RDF Layer* provides direct access to the non-normalized RDF data. The *Normalization Layer* expresses normalizations by means of RDF2RDF views, possibly supported by functional indices and caches. These indices and caches are used to improve the query performance against the normalized RDF data. Thereby, there are two kinds of views: Term-based views for crafting RDF terms, especially IRIs, and triple-based views for relating the terms to each other. If required the *Enrichment Layer* can be used to trigger the additional generation of information about existing entities. Finally the *Query Federation Layer* can be used to integrate other (possibly virtual) SPARQL endpoints. Note, that changes made in one layer are automatically propagated to the upper layers.

3 Discussion and Future Work

The authors are aware that RDF views come with performance implications [2]. It can be compared to a debug executable which executes slower by an order of magnitude compared to a release version. But this executable offers a lot of features which support the debugging process. In the same way the proposed design allows unique knowledge graph debugging capabilities. For some use cases the performance of this design might be sufficient and would not require a full RDF export. If a "release" version of the dataset is required, SPARQL queries can be used to retrieve portions of the data on-the-fly as well as export all data at once. This "release" data can then be loaded into a graph backend. This generic architecture can be used to add different data sources to the knowledge graph. These data sources could be databases, REST APIs, etc. Since a lot of company data is only accessible via REST APIs or via databases, this opens new integration opportunities for company data. Furthermore it would be possible to add a *Meta Data Layer* which could store provenance, confidence and other meta-information which is relevant for relations and entities. In addition it would be possible to add a *Fusion Layer* which would use all *owl:SameAs* relationships and entity fusion algorithms to show a fused view of all entities for the imported source datasets. Despite a possible performance loss by using RDF views, the advantages of such a design surpass the disadvantages. Hence we believe that this novel design will help to create knowledge graphs for many different heterogeneous data sources.

Acknowledgments. This work was supported by grants from the Federal Ministry for Economic Affairs and Energy of Germany (BMWi) for the SmartDataWeb Project (GA-01MD15010B)[13]

References

1. W. Beek, L. Rietveld, H. R. Bazoobandi, J. Wielemaker, and S. Schlobach. Lod laundromat: A uniform way of publishing other people's dirty data. In *ISWC*, 2014.
2. N. Konstantinou, D. E. Spanos, and N. Mitrou. Transient and Persistent RDF Views over Relational Databases in the Context of Digital Repositories. *Communications in Computer and Information Science*, 390 CCIS:342–354, 2013.

[13] `http://smartdataweb.de/`