Publishing Danish Agricultural Government Data as Semantic Web Data

Alex B. Andersen, Nurefşan Gür, Katja Hose,
Kim A. Jakobsen, and Torben Bach Pedersen

Department of Computer Science, Aalborg University, Denmark
{aban09,nurefsan,kjakob09,khose,tbp}@cs.aau.dk

Abstract. Recent advances in Semantic Web technologies have led to a growing popularity of the Linked Open Data movement. Only recently, the Danish government has joined the movement and published several datasets as Open Data. These raw datasets are difficult to process automatically and combine with other data sources on the Web. Hence, our goal is to convert such data into RDF and make it available to a broader range of users and applications as Linked Open Data. In this paper, we discuss our experiences based on the particularly interesting use case of agricultural data as agriculture is one of the most important industries in Denmark. We describe the process of converting the data and discuss the particular problems that we encountered with respect to the considered datasets. We additionally evaluate our result based on several queries that could not be answers based on existing sources before.

1 Introduction

In recent years, more and more structured data has become available on the Web, driven by the increasing popularity of both the Semantic Web and Open Data movement, which aim at making data publicly available and free of charge. Several governments have been driving forces of the Open Data movement, most prominently `data.gov.uk` (UK) and `data.gov` (USA), which publish Open Data from departments and agencies in the areas of agriculture, health, education, employment, transport, education, etc. The goal is to enable collaboration, advanced technologies, and applications that would otherwise be impossible or very expensive, thus inspiring new services and companies. Especially for governments, it is important to inspire novel applications, which will eventually increase the wealth and prosperity of the country. While publication of raw data is a substantial progress, the difficulty in interpreting the data as well as the heterogeneity of publication formats, such as spreadsheets, relational database dumps, and XML files, represent major obstacles that need to be overcome [9, 12, 15] – especially because the schema is rarely well documented and explained for non-experts. Furthermore, it is not possible to evaluate queries over one or multiple of these datasets.

The Linked (Open) Data movement (`http://linkeddata.org/`) encourages the publication of data following the Web standards along with *links* to other data sources providing semantic context to enable easy access and interpretation of structured data on the Web. Hence, publishing data as Linked Data (LD) [7, 8] entails the usage of certain standards such as HTTP, RDF, and SPARQL as well as HTTP URIs

as entity identifiers that can be dereferenced, making LD easily accessible on the Web. RDF allows formulating statements about resources, each statement consists of subject, predicate, and object – referred to as a triple. Extending the dataset and adding new data is very convenient due to the self-describing nature of RDF and its flexibility.

In late 2012, the Danish government joined the Open Data movement by making several raw digital datasets [3] freely available. Amon others, these datasets cover transport, tourism, fishery, companies, forestry, and agriculture. To the best of our knowledge, they are currently only available in their raw formats and have not yet been converted to LD. We choose agriculture as a use case, as it is one of the main sectors in Denmark, with 66% of Denmark's land surface being farmland[1]. Thus, there is significant potential in providing free access to such data and enabling efficient answering of *sophisticated* queries over it.

In this paper, we show how we made Danish governmental Open Data available as LD and evaluate the challenges in doing so. Our approach is to transform the agricultural datasets into RDF and add explicit relationships among them using links. Furthermore, we integrate the agricultural data with company information, thus enabling queries on new relationships not contained in the original data. This paper presents the process to transform and link the data as well as the challenges encountered and how they were met. It further discusses how these experiences can provide guidelines for similar projects. We developed our own ontology while still making use of existing ontologies whenever possible. A particular challenge is deriving spatial containment relationships not encoded in the original datasets. For a detailed discussion about the whole process, we refer the reader to the extended version of this paper [2]. The resulting LOD datasets are accessible via a SPARQL endpoint (`http://extbi.lab.aau.dk/sparql`) as well as for download (`http://extbi.lab.aau.dk/`).

The remainder of this paper is structured as follows; Section 2 describes our use case datasets and discusses the main challenges. Then, Section 3 describes the process and its application to the use case. Section 4 evaluates alternative design choices, while Section 5 concludes and summarizes the paper.

2 Use Case

We have found the agricultural domain to be particularly interesting as it represents a non-trivial use case that covers spatial attributes and can be extended with temporal information. By combining the agricultural data with company data, we can process and answer queries that were not possible before as the original data was neither linked nor in a queryable format.

Late 2012, the Ministry of Food, Agriculture, and Fisheries of Denmark (FVM) (`http://en.fvm.dk/`) made geospatial data of all fields in Denmark freely available – henceforth we refer to this collection of data as *agricultural data*. This dataset combined with the *Central Company Registry (CVR) data (`http://cvr.dk/`)* about all Danish companies allows for evaluating queries about fields and the companies owning them. In total, we have converted 5 datasets provided by FVM and CVR into Linked Open Data. We downloaded the data on October 1, 2013 from FVM [10] and from CVR.

[1] `http://www.dst.dk/en/%20Statistik/emner/areal/arealanvendelse.aspx`

Agricultural Data. The agricultural data collection is available in Shape format [6], this means that each *Field*, *Field Block*, and *Organic Field* is described by several coordinate points forming a polygon.

Field. The Field dataset has 9 attributes and contains all registered fields in Denmark. In total, this dataset contains information about 641,081 fields.

Organic Field. This dataset has 12 attributes and contains information about 52,060 organic fields. The dataset has attributes that we can relate to the company data, i.e., the CVR attribute is unique for the owner of the field and references the CVR dataset that we explain below. The fieldBlockId attribute describes to which "Field Block" a field belongs to.

Field Block. The Field Block dataset has 12 attributes for 314,648 field blocks and contains a number of fields [11]. Field Blocks are used to calculate the funds the farmers receive in EU area support scheme.

Central Company Registry (CVR) Data. The CVR is the central registry of all Danish companies and provides its data in CSV format. There are two datasets available that we refer to as *Company* and *Participant*.

Company. This dataset has 59 attributes [5] and contains information, such as a company's name, contact details, business format, and activity, about 603,667 companies and 659,639 production units.

Participant. This dataset describes the relations that exist between a participant and a legal unit. A participant is a person or legal unit that is responsible for a legal unit in the company dataset, i.e., a participant is an owner of a company. The Participant dataset describes 359,929 participants with 7 attributes.

The use case data comes in different formats and contains only a few foreign keys. Further, there is little cross-reference and links between the datasets and no links to Web sources in general. Spatial relationships are even more difficult to represent in the data and querying data based on the available polygons is a complex problem. In particular, to enable queries that have not been possible before, we cleanse and link the (Organic) Field datasets to the Company dataset so that we can query fields and crops of companies related to agriculture. The particular challenges that we address are:

- Disparate data sources without common format
- Lack of unique identifiers to link different but related data sources
- Language (Danish)
- Lack of ontologies and their use

3 Data Annotation and Reconciliation

In this section, we outline the process that we followed to publish the datasets described in Section 2. The complete procedure with its main activities is depicted in Fig. 1. All data in the data repository undergoes an iterative integration process consisting of several main activities:

Import: Extract the data from the original sources
Analyze: Gain an understanding of the data and create an ontology
Refine: Refine the source data by cleansing it and converting it to RDF
Link: Link the data to internal and external data

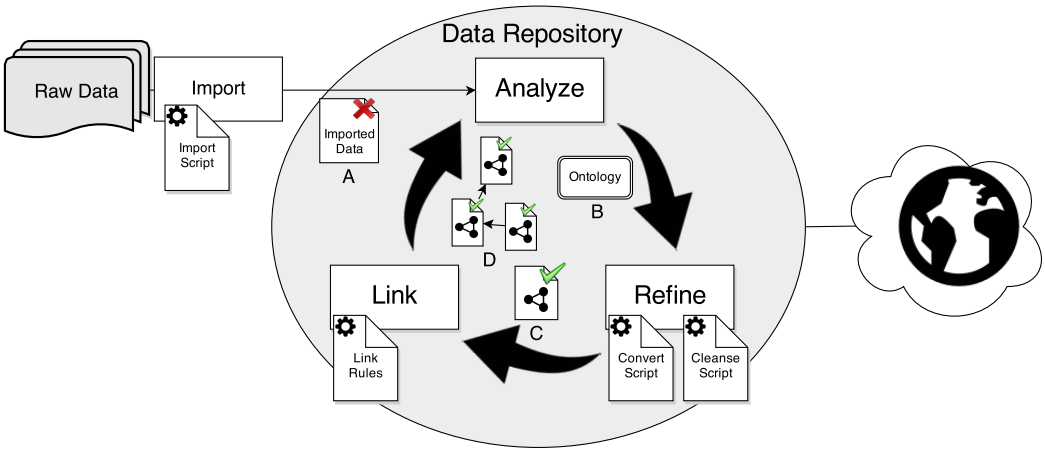

Fig. 1. Process overview

Data that has been through the integration process at least once may be published and thus become Linked Open Data that others can use and link to. In the remainder of this section, we will discuss these steps in more detail.

Import. The raw data is extracted from its original source into the repository and stored in a common format such that it is available for the later activities. The concrete method used for importing a dataset depends on the format of the raw data.

The agriculture datasets and CVR datasets introduced in Section 2 are available in Shape and CSV formats. Shape files are processed in ArcGIS[2] to compute the spatial joins of the fields and organic fields, thus creating foreign keys between the datasets. As the common format we use a relational database.

Analyze. The goal of this step is to acquire a deeper understanding of the data and formalize it as an ontology. As a result of our analysis we constructed a URI scheme for our use case data based on Linked Data Principles.

We strive to use existing ontologies as a base of our own ontologies. To do this, we make use of predicates such as `rdfs:subClassOf`, `rdfs:subPropertyOf`, and `owl:equivalentClass`, which can link our classes and properties to known ontologies.

Fig. 2 provides an overview of the ontology that we developed for our use case with all classes and properties. All arrows are annotated with predicates. The arrows with black tips represent relations between the data instances. The arrows with white tips represent relations between the classes. In short, we designed the ontology such that a Field is contained within a Field Block, which is expressed with the property `agri:contains` and is determined by a spatial join of the data. Organic Field is a subclass of Field and therefore transitively connects Field to Company. Field is also defined as being equivalent to the UN's definition of European fields from the AGROVOC [14] vocabulary. In addition we make use of other external ontologies and vocabularies, such as *GeoNames* [16], *WGS84* [4], and *FOAF (Friend of a Friend)* [1].

[2] `http://www.esri.com/software/arcgis`

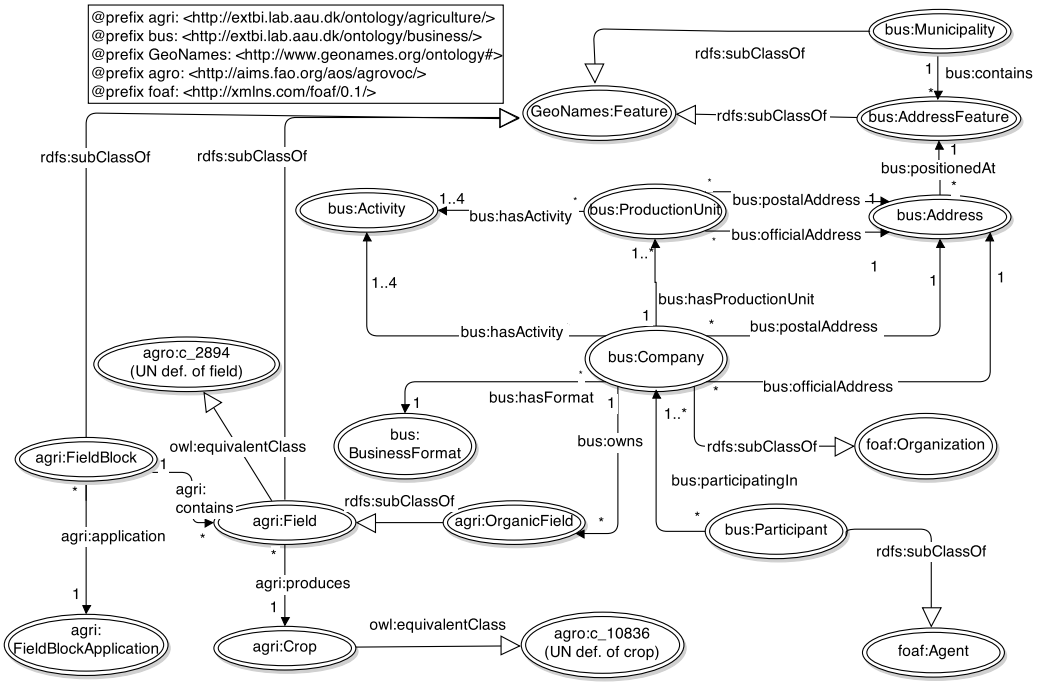

Fig. 2. Overview of the ontology for our use case

Refine. The Refine activity is based on the understanding gained in the Analyze activity and consists of data cleansing and conversion. Fig. 1 illustrates the data cleansing process where imported data and ontologies are used to produce cleansed data.

In our use case, we implemented data cleansing by using views that filter out inconsistent data as well as correct invalid attribute values, inconsistent strings, and invalid coordinates. Then we use Virtuoso Opensource [13] mappings to generate RDF data.

Link. The Link activity consists of two steps: internal linking and external linking, which converts the refined data into integrated data. The Link activity materializes the relationships between concepts and classes identified in the Analyze activity as triples. The example below shows our internal linking of the Field and the Field Block classes using the `geonames:contains` predicate.

```
agri:contains rdf:type owl:ObjectProperty ;
        rdfs:domain agri:FieldBlock ;
        rdfs:range agri:Field ;
        rdfs:subPropertyOf geonames:contains .
```

External linking involves linking to remote sources on instance and ontology level. On the ontology level, this means inserting triples using predicates such as `rdfs:subClassOf`, `rdfs:subPropertyOf`, and `owl:equivalentClass` that link URIs from our local ontology to URIs from remote sources. On instance level, we link places mentioned in the CVR data to equivalent places in GeoNames [16] using triples with the `owl:sameAs` predicate as illustrated in Fig. 3.

The overall process has provided us with analyzed, refined, and linked data; in total *32,457,657* triples were created. The result of completing this process is published

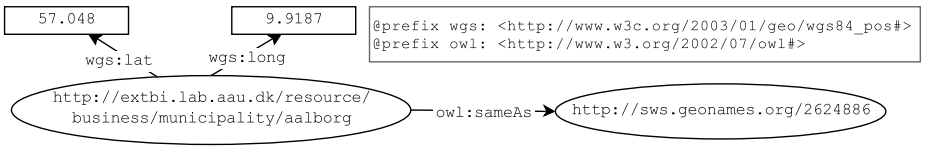

Fig. 3. External linking on instance level

and registered on `datahub.io`[3]. In case we wish to integrate additional sources, we simply have to reiterate through the process.

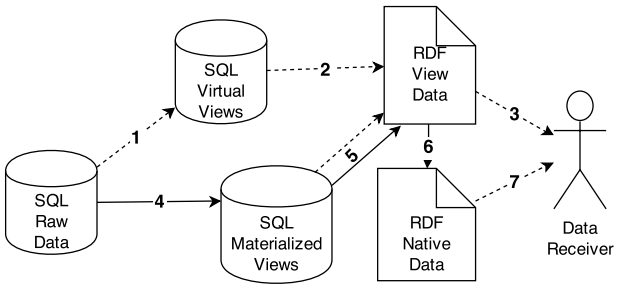

Fig. 4. Data flow for the materialization strategies

4 Experiments

In the following, we first describe three alternative design choices in the materialization of the data. They represent trade-offs between data load time and query time. We then discuss the results of our experimental evaluation, for which we ran an OpenLink Virtuoso 07.00.3203 server on a 3.4 GHz Intel Core i7-2600 processor with 8 GB RAM operated by Windows 7 Enterprise 64-bit SP 1.

The materialization strategies that we have considered are: *Virtual*, *relational materialization*, and *native*. Fig. 4 shows the different paths that data is traveling on; starting as raw data and ending at the user who issued a query. The solid lines represent data flow during the integration process whereas dashed lines represent data flow at query time.

Virtual. In the virtual strategy we perform data cleansing based on SQL views in the relational database. RDF mappings are formulated on top of these cleansing views to make the data accessible as RDF. To increase performance, we create a number of indexes on primary keys, foreign keys, and spatial attributes. In Fig. 4, using this strategy data flows through the arrows marked with 1, 2, and 3 at query time.

Relational Materialization. Here we materialize the above mentioned SQL views as relational tables. We create similar indexes as above but on the obtained tables. In Fig. 4, data flows through arrows 4, 5, and 3 – with 4 during load time and 3 and 5 during query time.

Native RDF. In this strategy, we extract all RDF triples from the materialized views and mappings and load them into a triple store. In Fig. 4, data flows through arrows 4, 5, and 6 during load time and arrow 7 during query time.

[3] `http://datahub.io/dataset/govagribus-denmark`

To test our setup, we created a number of query templates that we can instantiate with different entities and that are based on insights in agricultural contracting gained from field experts. Some of them contained aggregation and grouping (Aggregate Query Templates, AQT) others only standard SPARQL 1.0 constructs (Standard Query Templates, SQT).

For the virtual and relational materialization strategies we measured the load times for each step during loading – the results are shown in Table 1. Table 2 shows the execution times for our query templates on the three materialization strategies. Queries that run into a timeout are marked by a dash. As we can see, the native RDF strategy is faster than the two others, and relational materialized is generally faster than virtual. There is obviously a notable overhead when using views and mappings. On the other hand, the virtual strategy has very fast load time compared to the other strategies since no data has to be moved or extracted – in fact, the cleansing is delayed until query time. The relational materialized strategy is one order of magnitude faster in load time than the native strategy as it has less overhead during loading.

We can therefore conclude that the virtual strategy is well suited for rapidly changing data as it has minimum load time, the materialized strategy represents a trade-off between load time and query time and is suitable for data with low update rates, and the native strategy decouples RDF data from the relational data and is very suitable for static data.

Step	**Virtual**	**Materialized**	**Native**
Data Cleansing	74.92	603.35	603.35
Load Ontology	1.01	1.01	1.01
Load Mappings	8.76	12.35	12.35
Dump RDF	0.00	0.00	4684.82
Load RDF	0.00	0.00	840.04
Total	84.68	616.70	6141.56

Table 1. Load times in seconds

Query	**Virtual**	**Materialized**	**Native**
AQT 1	5.92	3.39	1.04
AQT 2	13.32	7.00	0.23
AQT 3	10.81	7.70	0.05
AQT 4	–	–	0.14
AQT 5	–	20.37	0.86
SQT 1	–	–	2.35
SQT 2	0.09	0.12	0.10
SQT 3	2188.85	1.81	0.40
SQT 4	6.57	2.35	1.63
SQT 5	–	23.79	3.29
Average	370.93	8.31	1.01

Table 2. Runtimes in seconds

5 Conclusion

Motivated by the increasing popularity of both the Semantic Web and the Open (Government) Data movement as well as the recent availability of interesting open government data in Denmark, this paper investigated how to make Danish agricultural data available as Linked Open Data. We chose the most interesting agricultural datasets among a range of options, transformed them into RDF format, and created explicit links between those datasets by matching them on a spatial level. Furthermore, the agricultural data was integrated with data from the central company registry. All these additional links enable queries that were not possible directly on the original data. The paper presents best practices and a process for transforming and linking the data. It also discusses the challenges encountered and how they were met. As a result, we not only obtained an RDF dataset but also a new ontology that also makes use of existing

ontologies. A particularly interesting challenge was how to derive spatial containment relationships not contained in the original datasets because existing standards and tools do not provide sufficient support. The resulting LOD datasets were made available for download and as a SPARQL endpoint.

Acknowledgment. This research was partially funded by "The Erasmus Mundus Joint Doctorate in Information Technologies for Business Intelligence – Doctoral College (IT4BI-DC)".

References

1. The Friend of a Friend (FOAF) Project. http://www.foaf-project.org/.
2. Alex B. Andersen, Nurefsan Gür, Katja Hose, Kim A. Jakobsen, and Torben B. Pedersen. Publishing Danish Agricultural Government Data as Semantic Web Data. Technical Report TR-35, 2014. http://dbtr.cs.aau.dk/DBPublications/DBTR-35.pdf.
3. Jakob Bøving Arendt. Denmark releases its digital raw material. http://uk.fm.dk/news/press-releases/2012/10/denmark-releases-its-digital-raw-material/, Ministry of Finance of Denmark, October 2012.
4. W3C-Dan Brickley. W3C Semantic Web Interest Group: Geo. http://www.w3.org/2003/01/geo/wgs84_pos, www.wgs84.com.
5. Erhvervsstyrelsen. Record layout: Juridiske enheder og P-enheder. http://www.cvr.dk/Site/Resources/Files/Media/RecordlayoutABO110.pdf.
6. ESRI. Shapefile technical description. *An ESRI White Paper*, 1998. http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf.
7. Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.
8. Tim Berners Lee. Design issues. http://www.w3.org/DesignIssues/LinkedData.html, July 2006.
9. Fadi Maali, Richard Cyganiak, and Vassilios Peristeras. A publishing pipeline for linked government data. In *The Semantic Web: Research and Applications*, pages 778–792. 2012.
10. Agriculture Ministry of Food and Fisheries of Denmark. FVM Geodata Download. https://kortdata.fvm.dk/download/index.html.
11. Danish Ministry of the Environment. Markblokkort (datasæt). http://www.geodata-info.dk/Portal/ShowMetadata.aspx?id=1eb89ebb-f674-4ad1-9e53-d1e252226596.
12. Martin G. Skjæveland, Espen H. Lian, and Ian Horrocks. Publishing the Norwegian Petroleum Directorate's FactPages as Semantic Web Data. In *ISWC*, pages 162–177, 2013.
13. OpenLink Software. Virtuoso RDF Views – Getting Started Guide, June 2007. http://www.openlinksw.co.uk/virtuoso/Whitepapers/pdf/Virtuoso_SQL_to_RDF_Mapping.pdf.
14. Agricultural Information Management Standards. AGROVOC Linked Open Data. http://aims.fao.org/aos/agrovoc/.
15. Boris Villazón-Terrazas, Luis.M. Vilches-Blázquez, Oscar Corcho, and Asunción Gómez-Pérez. Methodological Guidelines for Publishing Government Linked Data. In *Linking Government Data*, pages 27–49. Springer New York, 2011.
16. Marc Wick. GeoNames Ontology. http://www.geonames.org/ontology/documentation.html.