From Flat Lists to Taxonomies: Bottom-up Concept Scheme Generation in Linked Statistical Data

Albert Meroño-Peñuela[1,2], Ashkan Ashkpour[3] and Christophe Guéret[2]

[1]Department of Computer Science, VU University Amsterdam, NL
albert.merono@vu.nl
[2]Data Archiving and Networked Services, KNAW, NL
[3]International Institute of Social History, NL

Abstract. RDF Data Cube allows the modeling and publishing of Linked Statistical Data (LSD) in the Semantic Web. Often, variable values of such statistical data come in a non-standardized way and represented by too narrow, concrete or wrongly typed literals. Generally, adequate and standard *concept schemes* for such variables (especially in very specific domains like historical religious denominations, or building types in the pre-industrial era) do not exist and need to be created. This is a manual task that requires lots of expert knowledge and time investment. We present a workflow that combines hierarchical clustering and semantic tagging to automatically build concept schemes in a data-driven and bottom-up way, leveraging lexical and semantic properties of the non-standard dimension values. We apply our workflow in two different use-cases and discuss its usefulness, limitations and possible improvements.

Keywords. Linked Statistical Data, Standardization, Taxonomies, Clustering

1 Introduction

Motivation. RDF Data Cube is the standard for publishing multidimensional data in the Semantic Web, allowing linkage to other concepts and datasets in the so-called Linked Statistical Data (LSD) cloud. Statistical datasets often come in non-standardized ways, making it difficult to deal with comparability: a variety of dimension and value choices make it necessary to clean, correct and standardize the data before working with it. A common standardization practice is the creation and use of concept schemes. Some current concept schemes standardize common statistical concepts [10]. However, the lack of standard concept schemes in other domains is a great bottleneck for LSD publishers that wish to leverage (and ensure) reusability of concept scheme knowledge. Consequently, users are confronted with manual procedures in order to put dimension values (or codes) into meaningful groups.

Problem statement. Currently available concept schemes are not sufficient in order to standardize LSD. Researchers have to deal with unstructured and non-standardized dimension values, and are moreover confronted with 'information deluge', making the process of putting things into meaningful groups extremely complicated and time consuming, even with expert knowledge. Tools aiding this concept scheme building process for already published LSD are highly needed and non-existent, hampering the comparability and use of statistical datasets across the Web.

Use-cases. To enhance comparability studies in social history, researchers have been studying the dimension values of several historical LSD datasets. They have great interest in proposing concept schemes for the standardization of historical religious denominations and historical housing types.

Contribution. We propose a highly reproducible, generalizable and scalable workflow to automatically generate standard concept schemes from non-standardized dimension values in LSD datasets in a bottom-up way. We leverage the intrinsic lexical and semantic properties to propose meaningful classifications.

Findings. We find that the combination of lexical hierarchical clustering and semantic tagging of non-standard dimension values of our workflow provides useful support to the knowledge expert in the concept scheme building process.

The rest of the paper is organized as follows. We survey the related work in Section 2. In Section 3 we propose a workflow to automatically construct concept schemes from flat literals of non-standard values of dimensions in LSD. In Section 4 we present our experiments in Linked Census Data, before we conclude in Section 5.

2 Related Work

When working with non-standardized statistical data, the process of creating classification systems has been a mostly manual job. Current classification practices are therefore based mainly on data-driven, bottom-up, manual efforts by domain experts [1]. Researchers which lack programming skills, budget or sometimes necessitated by the data itself are bound to use (a combination of) different tools in order to clean, filter, group and classify statistical data before its publication: this is the purpose of OpenRefine [7]. A set of clustering algorithms (defined as "finding groups of different values that might be alternative representations of the same thing"[1]) are provided. Perhaps [8] is the closest match to the taxonomical knowledge construction via hierarchical clustering that we aim at, although fundamental differences apply with respect to the input data (collections of documents instead of flat literal lists) of different domains. Unfortunately, there is hardly any tool support available for conducting this standardization: (a) in a purely Linked Data setting; and (b) standardizing values *after* their publication as LOD in order to preserve both original and standard values.

3 Bottom-up Construction of Concept Schemes

We propose a workflow to automatically build bottom-up concept schemes from flat lists of non-standardized dimension values in LSD. We aim at RDF Data Cubes that need to preserve faithful representations of original source data: in such datasets, it is not possible to standardize dimension values *before* converting to RDF Data Cubes. The process is divided in five steps: retrieval of literals, hierarchical clustering, semantic tagging, linking and serializing.

[1] `https://github.com/OpenRefine/OpenRefine/wiki/Clustering-In-Depth`

3.1 Retrieval of Literals

First, literals of the non-standardized dimension values need to be retrieved. Since we are interested in building concept schemes in LSD, we use SPARQL queries that follow the template shown in Figure 1 against RDF Data Cube [4] datasets. Once executed, the resultset contains a list of unique non-standard dimension value literals.

```
PREFIX qb: <http://purl.org/linked-data/cube#>
PREFIX skos: http://www.w3.org/2004/02/skos/core#
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?dimValue ?literal
WHERE {
 ?obs a qb:Observation;
      eg:my-dim ?dimValue .
 ?dimValue skos:prefLabel|rdfs:label ?literal . }
```

Fig. 1. SPARQL template to retrieve non-standardized dimension value literals in RDF Data Cube datasets. `eg:my-dim` represents any entity of type `qb:DimensionProperty`.

3.2 Hierarchical Clustering

Our hypothesis is that knowledge experts group disparate literals mostly on a string similarity basis. Obviously, some literals may be grouped together for other reasons (e.g. semantic similarity), and it is part of our study to understand which ratio of the target concept scheme can be reached using lexical criteria only.

Since concept schemes are taxonomies, we choose *hierarchical clustering* as our method to build taxonomic relations between non-standard literals. We use the resultset of the previous step as input for the hierarchical clustering algorithm included in SciPy [3], and we use the Levenshtein edit distance [2] as a distance metric.

3.3 Semantic Tagging

An important task knowledge experts do when they build concept schemes is to label upper categories (e.g. the cluster containing “Barracks”, “Arsenal”, and “Citadel” may be named “Military buildings”). We suggest meaningful names for the output clusters of the previous step by leveraging semantic resources like WordNet and DBpedia. Concretely, we offer two alternatives for semantic tagging of clusters:

1. **Term-based tagging**. After the removal of stop words, we tokenize and stem all literals under the same cluster and rank them according to their appearance frequency. We use the top-1 token to query WordNet and DBpedia to get all of its synset and `skos:Concept`, respectively. We use those as suggestions to name the cluster.
2. **Bag-of-words tagging**. After the removal of stop words, we tokenize and stem all literals under the same cluster. We query WordNet and DBpedia using all tokens of all literals of the cluster, getting their synset and `skos:Concept`. We leverage `skos:broader` relations to find the closest common broader concept of all literals, and we use this concept as suggestion to name the cluster.

We consider all the descendant links below a cluster node *k* to belong to the same cluster if *k* is the first node below the cut threshold *t*. We use `t = 0.7 * max(d(k, i))`, where *d(k,i)* is the distance between the node *k* and any other node *i*.

3.4 Linking

After producing the concept scheme, we still need to link it to the original non-standard values. Since we have preserved the URIs of the original dimension values (see Figure 1), issuing links between the two is an almost trivial task.

3.5 Serializing

Once we have produced the concept scheme and the links back to the original dimension values, we serialize both datasets using SKOS [5] and RDF Data Cube [4], producing URIs for all new concepts.

4 Experiments with Linked Census Data

We use the workflow proposed in Section 3 to build bottom-up classifications of non-standard dimension values in the RDF Data Cube version of the Dutch Historical Censuses dataset (CEDAR)[2]. This dataset is produced by TabLinker[3], converting Excel census tables to RDF Data Cube [6]. We use our approach to build classification schemes on top of non-standard dimension values in this dataset[4].

4.1 Input Data

The CEDAR dataset covers a long time period (1795-1971) in which lots of *religious denominations* were registered in a non-standard way. We aim at producing a standard concept scheme to cover all these religious denominations[5]. Similarly, the census also registered counts for houses, encoding a dimension *house type* in a non-standard way. We aim at producing a standard concept scheme to cover all housing types[6]. Our gold standards are classification schemes developed by knowledge experts on top of the Dutch historical censuses. For historical religions, researchers manually standardized and coded the variables belonging to the same religious denomination[7]. The final outcome is a classification system of historical religions containing 210 unique denominations. For historical house types, we use another expert-based classification based on a manual and straightforward approach in which the terms are classified according to their functions[8]. We compare the results of our proposed workflow with these expert-crafted classifications.

[2] See `http://www.cedar-project.nl`

[3] See `https://github.com/Data2Semantics/TabLinker/`

[4] See `https://github.com/CEDAR-project/TabCluster/`

[5] See `http://goo.gl/PSmIzy` for the input religions

[6] See `http://goo.gl/Hsqwz0` for the input house types

[7] See `http://goo.gl/qT2vIX` for the expert-based classification system of religions

[8] See `http://goo.gl/mt1dsn` for the expert-based classification system of house types

4.2 Results and Discussion

We execute several times our workflow on the input datasets, exploring appropriate parameter values for hierarchical clustering. We take the average term distance to determine the distance between clusters. Figure 2 shows our resulting schemes[9].

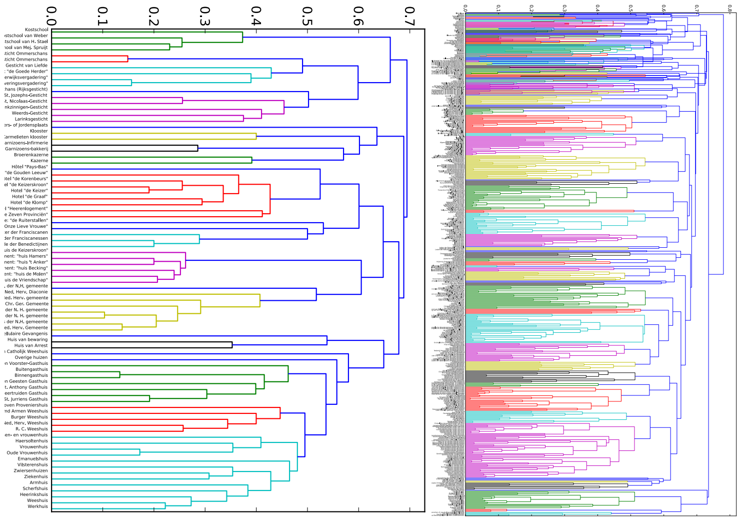

Fig. 2. Dendograms of the hierarchical clusters of CEDAR house types and historical religions. See `http://goo.gl/ggMqiR` and `http://goo.gl/VG9IGU` for full resolution images.

We observe interesting groups being identified in the building types dataset. For instance, the cluster containing the values "Klooster der Franciscanen", "Klooster van de orde der Franciscanessen" and "Klooster van de orde der Benedictijnen" clearly identifies *kloosters* (monasteries), and gets appropriately `http://nl.dbpedia.org/resource/Klooster` (and its English equivalent `http://dbpedia.org/resource/Monastery`) as a semantic tag for the broader category of the concept scheme. Similarly, instances of historical religions that identify "Apostolic" or "Protestant" denominations are grouped together under the same cluster. Interestingly, a purely lexical approach exploits the transitivity of some string similarities (e.g. "Kazerne" and "Militair Ziekenhuis" are clustered together due to the linking member "Militair Kazerne" of the same cluster). On the other hand, the purely lexical clustering shows also its limitations when instances like "ziekenhuis" (hospital), "armhuis" (poorhouse) or "weeshuis" (orphanage) are clustered together (due their common suffix "-huis") despite their notable semantic differences.

Knowledge experts validating our workflow compare these results with the gold standards, and see its usefulness when building concept schemes to standardize historical statistical data. Concretely, they are interested in its application as a knowledge

[9] See `https://github.com/CEDAR-project/TabCluster/` for algorithm output.

support tool in the concept scheme building process. Accordingly, a key issue of the process, covered by our workflow, is leveraging the combination of lexical and semantic structuring. Experts truly think that a combination of both approaches is what indeed goes on when they execute the process manually. It is to be seen, though, the trustworthiness of our proposed workflow as a totally autonomous tool.

5 Conclusions and Future Work

In this paper we present an automatic approach to generate concept schemes from non-standard dimension values in Linked Statistical Data. We propose a workflow that combines hierarchical clustering to leverage lexical relatedness, with the enrichment from external knowledge bases to leverage semantic relatedness. As a result, we produce concept schemes that knowledge experts can compare with their manually generated ones. We plan to extend this work in multiple ways. First, we will systematically compare our workflow output with the gold standards, in order to get precision/recall scores that evaluate our approach. Second, we will execute the workflow against arbitrary datasets, to confirm its domain independence. Third, we plan on finding optimal values of the *t* threshold, here set by empirical exploration. Finally, we will generalize our proposal by implementing additional clustering algorithms (e.g. Latent Semantic Analysis) and other semantic methods for cluster tagging.

Acknowledgements. This work was supported by the Computational Humanities Programme of the KNAW (see `http://ehumanities.nl`) and the Dutch national program COMMIT.

6 References

1. Esteve, A., Sobek, M. Challenges and Methods of International Census Harmonization. Historical Methods, 36(2):37-41, 2003.
2. Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10(8):707-710, 1966.
3. Jones, E., Oliphant, T., Peterson, P. et al. SciPy: Open source scientific tools for Python. `http://www.scipy.org/` (2001).
4. The RDF Data Cube Vocabulary, World Wide Web Consortium. `http://www.w3.org/TR/vocab-data-cube/` (2014)
5. SKOS Simple Knowledge Organization System Reference, World Wide Web Consortium. `http://www.w3.org/TR/2009/REC-skos-reference-20090818/` (2009)
6. Meroño-Peñuela, A., Ashkpour, A., Rietveld, L., Hoekstra, R., Schlobach, S. Linked Humanities Data: The Next Frontier? A Case-study in Historical Census Data. Linked Science workshop, ISWC (2012).
7. Huynh, D., Mazzocchi, S. OpenRefine. `http://openrefine.org`
8. Knijff, J. de, Frasincar, F., Hoogenboom, F. Domain taxonomy learning from text: The subsumption method versus hierarchical clustering. Data & Knowledge Engineering, 83, pp. 54-69 (2013)
9. Meroño-Peñuela, A., Ashkpour, A., van Erp, M., Mandemakers, K., Breure, L., Scharnhorst, A., Schlobach, S., van Harmelen, F. Semantic Technologies for Historical Research: A Survey. Semantic Web Journal (to appear) (2012)
10. SDMX Content Oriented Guidelines, `http://sdmx.org/?page_id=11`