Towards Budget Comparative Analysis: The Need for Fiscal Code Lists as Linked Data

Panagiotis-Marios Filippidis
Open Knowledge Greece and Aristotle University of Thessaloniki
School of Journalism and Mass Communications
Thessaloniki, Greece
pafilipp@jour.auth.gr

Sotirios Karampatakis
Open Knowledge Greece and Aristotle University of Thessaloniki
School of Mathematics
Thessaloniki, Greece
sokaramp@auth.gr

Lazaros Ioannidis
Open Knowledge Greece and Aristotle University of Thessaloniki
School of Mathematics
Thessaloniki, Greece
larjohn@math.auth.gr

Jindřich Mynarz
Department of Information and Knowledge Engineering, University of Economics
W. Churchill Sq. 4
130 67 Prague, Czech Republic
jindrich.mynarz@vse.cz

Vojtěch Svátek
Department of Information and Knowledge Engineering, University of Economics
W. Churchill Sq. 4
130 67 Prague, Czech Republic
svatek@vse.cz

Charalampos Bratsas
Open Knowledge Greece and Aristotle University of Thessaloniki
School of Mathematics
Thessaloniki, Greece
cbratsas@math.auth.gr

ABSTRACT

Code lists are a key part of budget datasets as they serve for the coding of fiscal concepts within them. However, the great diversity of classifications across countries and concepts does not allow to presume upon their actual value, as dimension properties. In this paper we discuss the need for creating code lists Linked Data for the classifications used in fiscal datasets, in three basic steps. First, code lists have to be extracted from fiscal datasets, especially if there are no relevant metadata in the budget description, which could easily identify them. Next, code lists from different datasets or sources have to be represented in the same way, with SKOS vocabulary, thus they can be linked with each other. Finally, linking of similar code lists will also allow the linking of the containing datasets, increasing their data analysis and knowledge extraction possibilities.

CCS Concepts

•**Information systems** → **Resource Description Framework (RDF); Ontologies; Extraction, transformation and loading;** *Hierarchical data models; Information retrieval diversity;*

Keywords

Linked Data, SKOS, Knowledge Extraction

1. INTRODUCTION

Budget datasets contain detailed information about the ways public money is spent to the functions of the government. They include fields that refer to fiscal and other budget related concepts. Many of these fields have a specific range of values. To this end, statistic agencies across Europe have created appropriate code lists, which are prescribed controlled vocabularies that contain all the values a specific field can get.

Code lists are an essential part of a budget dataset as they serve for the coding of concepts that can be written or described in many ways. The hierarchical structure of the majority of the code lists allows the hierarchy of concepts related to budget, thus they may support aggregated views over data, for example, over a particular expenditure category, or a municipality administration office. Additionally, standardized code lists are a key device to make fiscal data comparable. This information can be shared across several datasets and be interlinked to external data too, allowing comparisons between budgets of different years and different organizations as well, as they use the same codes for concepts that they would otherwise describe in a different way.

International authorities propose many generic code lists that have been fully or partially adapted to the national budget representation of European countries. The most common case of differentiation is when a national code list contain one or two additional levels of detail in relation to the international classification, according to the respective needs of the country. Furthermore, European countries have also established and use their own code lists. That leads to a variety of classifications for the same concepts, while they may use code lists in different fields of the budget, as well.

However, the use of officially proposed code lists in budgets is rather limited so far. We noticed that in many cases, the classifications used in budget datasets differ from both international and national proposed code lists and there are no relevant metadata about them. Thus, the ambiguous use of code lists by countries and municipalities is a very com-

SEMANTICS 2016: Posters and Demos Track
September 13-14, 2016, Leipzig, Germany

plicated issue that creates the need for their identification, extraction and linking.

The identification of code lists used by European authorities, is the first step towards their linking and subsequently, the linking of budget datasets of European countries and municipalities, via their common fields. The linking of different code lists is particularly important and requires their representation in a common vocabulary, like SKOS[1], in order to fully leverage their advantages. Then, SKOSified classifications that refer to the same concept can be connected in a automatic way.

In this way budget data will be useful for any citizen, as their transformation and linking will enable effective access to the included information. Nowadays a user has to browse large amounts of budget data in PDF, text or sheet format, requiring considerable time and effort to come to a conclusion. If the data is connected, knowledge retrieval techniques can automatically result all the relevant information in a user's search. This knowledge is valuable to every citizen and also to journalists and economic analysts as well, while governments and public authorities support the principles of open government, public access and transparency.

The main text of the paper is organized as follows: in Section 2 we discuss the extraction of code lists from budget datasets and its difficulties, while in Section 3 we present the need for a common representation and three tools that we have used in order to transform in SKOS main economic code lists. In Section 4, we examine their linking possibilities and finally, we conclude and list the next steps of our work in Sections 5 and 6, respectively.

2. EXTRACTION OF CODE LISTS

While code lists can be discovered within almost every budget dataset, their actual value, as hierarchical dimension properties, does not emerge, because many times the lists' integration into the dataset does not allow the user to extract information in order to use it for a purpose other than plain-text filtering. In other circumstances, code list terms are found within datasets not in a formal representation (i.e. a unique identifier), but with literals. This is common with countries, municipalities and other geographical attributes found in fiscal datasets.

The extraction of a code list from a fiscal dataset usually results in a flat list of distinct terms. A flat codes list may contain hierarchical relations information within the terms' names. Many times, such flat lists have already been published by their author, so the extraction process is more straightforward and involves downloading the list from a remote server and validating its contents against a format specification. The latter is necessary, in cases where the provided data is claimed to carry a specific format (for instance CSV), but contains syntax errors. To be useful at a later stage, the validated list can be transformed to a format that provides semantic relationships, initially among the terms of the same list. A semantically complete representation of the code list can then be used to generate any of the simpler representations that may require smaller amounts of information.

However, if the corresponding code list has not been published yet, which is often the case, its identification and extraction from a dataset can be a difficult process. Fiscal datasets may not include every code of the classifications and the completion of the missing parts and their hierarchic relations from relevant datasets (i.e. of another fiscal year) may be unattainable. Furthermore, if a European municipality uses its own code lists exclusively, the extraction effort needed is probably excessively bigger than its benefits, due to its limited use.

The most frequent problem though is the absence of the codes of the classification terms in the budget datasets, which usually contain just their literal value, without a complementary column denoting the corresponding code of the term. This did not allow us to identify many classifications in European fiscal datasets, especially in languages other than Greek or English. Thus, we extracted a few code lists, mainly from Greek budget datasets and we extended our methodology emphasizing in widely proposed by statistic agencies across Europe, classifications.

Generally, in order to achieve the code list extraction, we need to create a software system that takes as input a fiscal dataset and after processing it, it yields a set of code lists that are contained in the dataset. In this abstract definition we need to also add to the input information on which dataset columns consist of code list terms, and also information on the format of the code list as a whole.

A second software system would then be placed in front of the former system, to transform the distinct code list terms into a richer representation. The added data features can be hierarchical relationships between terms and additional attributes that the terms can have, usually coming from external sources. This process may also require integration with a separate system that takes care of linking the various code lists based on similarity of terms across datasets. The result of these two processes will be a semantic representation of objects, with each object matching a specific term and containing a label, a set of relationship with other terms and a set of additional attributes.

A third process could then create updated dataset copies, where the code list attributes are replaced by their respective semantically represented terms, using a unique identifier.

Finally, the extraction of code lists from fiscal datasets will allow to:

- Replace their literal representation with a more machine-readable one, which can be easier deduplicated.
- Organize the code list to an explicitly hierarchical format. Many code lists are already hierarchical but there are no documented links between terms, so the hierarchy is denoted by naming conventions only. A semantically complete representation would include these relations explicitly. The accuracy of this process depends on the data structure and the way the hierarchy is expressed within the codes, but it usually can be automated.
- Link the similar code lists into families and subsequently link the containing datasets, making comparisons and data analysis more straightforward. Even standardized code lists are often overridden in order to accommodate each state's specific needs, which in turn overlap with similar needs in other states.

3. TRANSFORMATION

The common representation of code lists is a sine qua non task for their linking. To this end, we transformed some of

[1]https://www.w3.org/TR/skos-reference/

the most common classifications in the economic field, into the SKOS vocabulary, using three different software tools.

3.1 Need for a common representation

Code lists from different datasets should have a similiar representation, thus they can be linked with each other. This makes easier to apply data mining and analysis techniques in fiscal datasets, in order to extract further and previously "hidden" knowledge within them. The most suitable way to do this is using the SKOS vocabulary, as it is expressed in RDF (inherently supporting localized labels), it is standardized and platform independent and allows for hierarchical ordering of terms.

But beyond the connection of datasets from different countries and municipalities, the necessity of a common representation for code lists arises from the little reuse code lists get in fiscal data and the lack of standardization in describing fiscal datasets, apart from EU-level code lists.

In particular, in many cases, there are many different code lists for the same field or concept in a country's datasets, i.e. between two different municipalities or two fiscal years of the same municipality. However, no metadata about the budget datasets and the classifications they include, are provided, by the publishing authorities. Additionally, the majority of the national (or international) classifications that statistic agencies propose can not be identified in the corresponding country's fiscal datasets. This kind of inconsistency and the lack of any information about code lists within the datasets requires their transformation and later, their linking to bridge the gaps between seemingly disparate classifications.

3.2 SKOSified Classifications

The code lists that we selected to transform in SKOS [3], based on their content and their popularity in international and national statistic agencies are:

- the Geographical Standard Code List (GEO)
- the Sectors Codes (CL_SECTOR)
- the Statistical Classification of Products by Activity, Version 2.1 (CPA 2.1) from Eurostat and the Central Products Classification (CPC) from UNSD
- the Organisation Identifier Code List from IATI
- main fiscal code lists from the European System of National and Regional Accounts (ESA 2010) such as: assets and liabilities, balancing items and net worth, distributive transactions, financial assets, institutional sectors, non-financial assets, other changes in assets, transactions in non-produced and non-financial assets and transactions in products
- the budget expenditure and budget revenue codes of Greece, Greek municipalities and Greek regions
- the administration offices of Greek municipalities and Greek regions

3.3 SKOSifying Tools

We used three software tools, OpenRefine[2], UnifiedViews[3] and LinkedPipes ETL[4] for processing code lists coming into various formats. OpenRefine accepts into its input a handful of data formats, ranging from CSV and Excel-like spreadsheets, to JSON and RDF. It also has an RDF extension that enables to build the SKOS model and export the data in RDF format. OpenRefine's strength is its rapid approach to the targeted SKOS output through a graphical interface which contains the RDF skeleton of the mapping (see Fig. Figure 1). OpenRefine is not able to repeat the same conversion process with a refreshed dataset, but this can be achieved via BatchRefine[5].

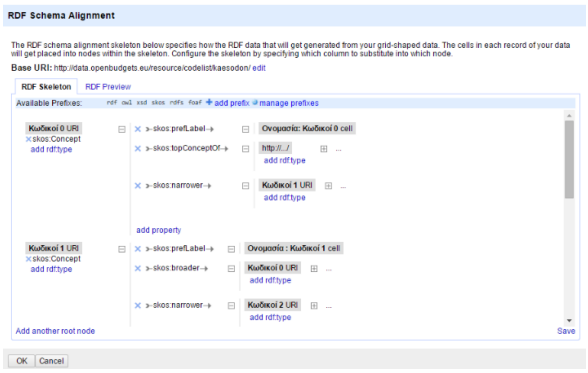

Figure 1: Open Refine RDF skeleton

On the other hand, UnifiedViews [2] and LinkedPipes ETL [1] provide more advanced options and support automated execution of the extraction and conversion pipelines. A components pipeline was designed in order to represent a set of scenarios and serve as the reference implementation for further needs on extraction and transformation of code lists. The reuse of existing components is crucial on the design and the development of such a pipeline.

However, the pipeline design required for each code list is highly dependent on the input format and the desired output. Different pipeline structure and configuration may need to take place for code lists of different structure or format.

A UnifiedViews/LinkedPipes ETL pipeline is composed from several data processing units (DPU), each one having a specific functionality. Many different DPUs have been included in our code list transformation pipelines for data input, cleaning, configuration, basic mappings, as well as complex mappings, using SPARQL queries for semantic relations between entities. An example of a UnifiedViews pipeline is shown in Figure 2.

LinkedPipes ETL offers even more possibilities and configuration features, as well as a user-friendy graphical interface. We used it alongside UnifiedViews, to SKOSify some latter code lists. No performance issues have been arised from using these tools, as they have been tested even for the RDF transformation of big budget datasets, beside code lists. The most important thing is their functionality and the features and options they provide, and a user may select any tool based on his data.

Totally, each of the aforementioned classifications were transformed into SKOS with these tools, and thus, was able to get easily linked to similar code lists. The SKOSified code lists, along with additional technical stuff such as their

[2]http://openrefine.org
[3]http://www.unifiedviews.eu/
[4]http://etl.linkedpipes.com/

[5]https://github.com/fusepoolP3/p3-batchrefine

pipelines are available on the Github Repository of Open Budgets EU[6].

Figure 2: Unified Views Pipeline

4. LINKING CODE LISTS

Linking code lists is a device of comparability. Links among the code lists ease their use in combination, because the links express explicitly how the code lists' concepts can be compared. In turn, the datasets described by these code lists become comparable as well. For example, if there is a `skos:exactMatch` link between code list concepts, they can be used interchangeably. In this way, querying across data becomes feasible and comparison of data points more meaningful.

However, a fundamental problem of linking code lists is that their concepts are typically not described with enough data to determine if the linked concepts match or not. It is often the case that a code list concept is described only with a code (`skos:notation`) and a label (`skos:prefLabel`) in a single language. Not only this is not enough to establish reliable links automatically, domain experts are frequently at a loss when asked if such concepts match. Faced with this uncertainty, the remaining option is to learn the concepts' similarity from the way they are used in data. However, the data described by the concepts may reveal that the users of the code lists assigning the concepts are confused by their ambiguity too.

Motivated by the need for data comparability, we created links between 16 pairs of datasets used in the OpenBudgets.eu project, amounting to the total of 20 975 links. Most of these links were discovered via reused codes that were typically reformatted into new codes or IRIs. We mainly used simple link discovery rules formalized as SPARQL 1.1 Update operations. The generated linksets along with linkage rules used to produce them are available on the Open Budgets EU Github Repository[7].

However, in many cases the automated approaches we employed produced partial alignments between the linked code lists. While some code list concepts are linked to their counterparts, others are left unlinked. Partial alignments are insufficient for complete data migrations, so their value lies instead mostly in allowing interactive discovery of relevant linked data. Since automated linking can rarely achieve complete coverage if data is scarce, there is a need for a semi-automated approach that involves human input.

[6]https://github.com/openbudgets/Code-lists
[7]https://github.com/openbudgets/linksets

5. CONCLUSION

Code lists are an essential part of fiscal datasets. Their extraction and transformation into a common vocabulary are vital processes for the effective discovery of information within budget data. Extracting specific classifications from fiscal datasets is a complicated task which in cases may be even impossible. The process of transformation can be automated in order to produce standard SKOS representation of the code lists. Three tools were tested for the purpose, with UnifiedViews and its successor, LinkedPipes qualifying as the more versatile, given the versatility of the datasets and their ability to let the user easily reuse the process as the code lists change. We demonstrate ways for automated linking of classifications, but this task could be very difficult to accomplish largely, so, semi-automated tools would improve the linking possibilities. Connecting as much as possible code lists highly increases the knowledge extraction potential from budget datasets and thus their publication actually fulfills its real purpose.

6. FUTURE WORK

The limited extraction possibilities of code lists from fiscal datasets denote the need for creating a backbone of connected code lists, to easily compare and identify new code lists in budget data. Apparently, transforming these classifications into SKOS increases their linking possibilities and this is a another part of our work that is not closed. The automated linking of code lists is a crucial first step to this direction and advanced tools supporting and guiding the linking procedure is our next goal. This will hopefully help us make more accurate links, in order to interconnect more code lists and subsequently, more fiscal datasets and all the information they include.

Acknowledgments

This work has been supported by the OpenBudgets.eu Horizon 2020 project (Grant Agreement 645833).

7. REFERENCES

[1] J. Klímek, P. Škoda, and M. Nečaský. LinkedPipes ETL: Evolved linked data preparation. In *The Semantic Web: ESWC 2016 Satellite Events - ESWC 2016 Satellite Events, Anissaras, Crete, Greece, May 29-June 2, 2016, Revised Selected Papers, to appear*, 2016.

[2] T. Knap, M. Kukhar, B. Macháč, P. Škoda, J. Tomeš, and J. Vojt. Unifiedviews: an etl framework for sustainable rdf data processing. In *European Semantic Web Conference*, pages 379–383. Springer, 2014.

[3] A. Miles, B. Matthews, M. Wilson, and D. Brickley. Skos core: simple knowledge organisation for the web. In *International Conference on Dublin Core and Metadata Applications*, pages pp–3, 2005.