ODSAG: Enhancing Open Data Discoverability and Understanding through Semantic Annotation

Abiola Paterne Chokki[1], Rabeb Abida[1], Benoît Frénay[1], Benoît Vanderose[1] and Anthony Cleve[1]

[1]University of Namur, Rue de Bruxelles 61, 5000 Namur, Belgium

Abstract

Many governments have published their data on the web with the goals of improving transparency and stimulating innovation, among others. In order to achieve these goals, users must be able to discover and understand these Open Government Data (OGD). The use of semantic annotation has been proven in previous studies to be effective in meeting this need. Yet, the process of annotating data remains an open challenge. Although efforts have been made in recent years to simplify this process, there is still a lack of semantic annotation tools that integrate well with OGD portals. To this end, we present ODSAG (Open Data Semantic Annotation and Graph), a chrome extension that can be easily interoperable with any OGD portal, automatically annotates an open dataset and creates graphs from it.

Keywords

Open Government Data, Discoverability, Understanding, Semantic Annotation, Knowledge Graph

1. Introduction

Around the world, many governments have implemented Open Government Data (OGD) policies to make their data more accessible and usable by the public [1]. The release of these data is most often motivated by values that include improving government transparency [2] and stimulating innovation [3, 4]. However, there is still a number of barriers that prevent various OGD initiatives from reaching their full potential [1]. Among these challenges are discoverability and understanding [5, 6, 7]. Indeed, before being able to use the data, consumers must be able to find data relevant to their needs (discoverability) [6]. Once they discover the relevant data, they must be able to understand the metadata and content of the data in order to exploit it, such as for data integration, data cleaning, data mining, machine learning and knowledge discovery tasks (understanding) [5, 7].

The use of semantic annotation and Linked Open Data (LOD) has been proven in previous studies [8, 9, 6] to be efficient in solving the mentioned challenges. Semantic annotation is the process of assigning semantic tags from Knowledge Graphs (KGs) (e.g., Wikidata, DBpedia) to data items [5]. It mainly consists of the following tasks (see Figure 1): (1) Column Type Annotation (CTA) which involves assigning a semantic type (e.g., a Wikidata class) to each

EGOV-CeDEM-ePart 2022, September 06–08, 2022, Linköping University, Sweden (Hybrid)

abiola-paterne.chokki@unamur.be (A. P. Chokki); rabeb.abida@unamur.be (R. Abida); benoit.frenay@unamur.be (B. Frénay); benoit.vanderose@unamur.be (B. Vanderose); anthony.cleve@unamur.be (A. Cleve)

0000-0003-4500-2141 (A. P. Chokki); 0000-0003-4005-2633 (R. Abida); 0000-0002-7859-2750 (B. Frénay); 0000-0001-9752-0085 (B. Vanderose); - (A. Cleve)

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3399/paper11.pdf

column (see green color), (2) Cell Entity Annotation (CEA) which involves mapping each cell to an entity in KG (see magenta color) and (3) Column Property Annotation (CPA) which involves assigning a property or predicate in KG to the relation between two columns (see blue color). Although the benefits of semantically annotated data have been widely recognized, there is still a vast number of datasets without any semantic annotation being published on open data portals every day, probably because adding semantic annotations to data is a laborious, error-prone, challenging task for publishers [8] and also because the majority of existing tools are not capable of automating the process and being both interoperable with open data portals. Therefore, our research question is as follows: "*How to design a tool that supports semantic annotation of OGD?*"

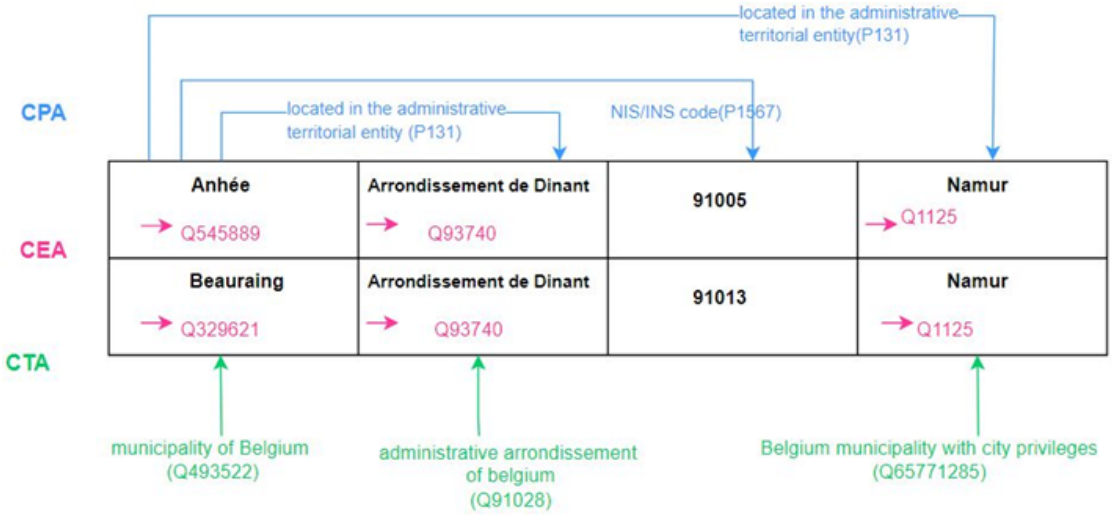


Figure 1: Semantic annotation using a COVID data[1] from the Namur (Belgium) open data portal. Tabular data (black) is annotated with properties (blue), entities (magenta), and types (green) using Wikidata

The methodology used to answer the research question is structured in three parts. First, we identify, through existing tools, a list of requirements that should be included in a tool that can facilitate the semantic annotation of OGD. Then, we implement these requirements in the tool called ODSAG **(Open Data Semantic Annotation and Graph)**. ODSAG has been implemented as a chrome extension that can be easily interoperable with any OGD portal, automatically annotates open data and creates graphs from the annotated data. Finally, we evaluate its effectiveness using a COVID dataset available on the Namur, Belgium portal. The target users of ODSAG are primarily publishers who can use the tool to annotate their data before publishing them, but can also be extended to data analysts who can use the tool to self-annotate unannotated datasets found on OGD portals to improve their understanding before using them.

The rest of this paper is divided into four main sections. Section 2 explores existing tools for semantic data annotation. Section 3 presents the requirements identified to support OGD semantic annotation. Section 4 describes the proposed ODSAG prototype. Section 5 describes

[1]https://rb.gy/xigrww

the implementation of ODSAG and demonstrates a use case. Finally, Section 6 provides a conclusion that summarizes the contributions of this paper and proposes some directions for future work.

2. Related Work

Open data on the web still has a high level of heterogeneity, lack of metadata and lack of interoperability, making it difficult to explore and understand. Unfortunately, many data producers are not familiar with LOD technologies and are not willing to invest time to integrate their data with KGs (semantic annotation). To address these gaps, many tools have been proposed. Reviewing each of them would beyond the scope of this paper. We have focused here on the most recent or most cited tools in the literature: OpenRefine [2], SemanticBot [10], Odalic [11], DataGraft [12], MantisTable [13], Mtab [14], JenTab [15]. Table 1 presents the reviewed tools according to the following criteria: annotation method (C1), data pre-processing capability (C2), subject-column detection capability (C3), semantic annotation (C4), technologies based on (C5), KGs or ontologies used (C6), graph generation capability (C7), export capability (C8) and interoperability capability with open data portals (C9). Table 1 also provides a comparison with the proposed tool, ODSAG.

Table 1
List of reviewd tools for semantic annotation. SA = Semi-Automatic, M = Manual, A = Automatic, N = No, Y =Yes and N/A = Not Applicable

Tools	Criteria										
	C1	C2	C3	C4			C5	C6	C7	C8	C9
				CTA	CEA	CPA					
SemanticBot	SA	N	Y	Y	N	Y	N/A	DBpedia YAGO LOV	Y	Y	Open Data Soft
OpenRefine	SA	Y	N	N	Y	Y	N/A	Wikidata	N	Y	N/A
Odalic	A	N	N	Y	Y	N	TableMiner+	Wikidata DBpedia	N	Y	N/A
DataGraft	M	N	N	Y	Y	N	Grafterizer		N	N	N/A
MantisTable	A	Y	Y	Y	Y	Y	N/A	Wikidata DBpedia	N	N	N/A
Mtab	A	Y	Y	Y	Y	Y	N/A	Wikidata DBpedia	N	N	N/A
JenTab	A	Y	Y	Y	Y	Y	N/A	Wikidata DBpedia	N	N	N/A
ODSAG	A	Y	Y	Y	Y	Y	Mantistable Mtab	Wikidata	Y	N	All

Referring to Table 1, none of the reviewed tools is able to be both interoperable with open data portals, automatically annotate data, and generate graph from the annotated data. This

[2]https://openrefine.org/

justifies the need for our ODSAG tool, which, compared to the other tools, can satisfy all these features.

3. Identification of Requirements to Support Semantic Annotation of Open Data

Based on the tools reviewed in Section 2, we are able to identify the list of requirements that a tool might have to support semantic annotation of open data. The identified requirements are a summary of existing features in the reviewed tools (see Table 2).

Table 2
List of requirements identified to support semantic annotation of open data

No	Requirements
R1	Facilitate the selection of data from OGD portals
R2	Provide a pre-processing of selected data
R3	Provide a subject column detection. This feature will allow to identify the main column and not to use the first column as the main column every time.
R4	Provide an automatic annotation of data (including CEA, CPA and CTA)
R5	Facilitate the export of the annotated data
R6	Generate a graph of the annotated data
R7	Easy to use and install

Once the requirements are identified, we implement them in a tool that we will present in the next section.

4. ODSAG Prototype

This section describes our proposed ODSAG prototype, which aims to address the shortcomings mentioned in the previously discussed tools and to incorporate the identified requirements in Section 3. Instead of starting from scratch, the ODSAG prototype integrates an existing automatic semantic table annotation tool Mtab [14] into its process and then enhances it (1) to make it interoperable with any open data portal and (2) to generate graphs from the annotated data. Figure 2 presents an overview of the prototype which consists of 4 steps: 1) dataset selection; 2) data pre-processing; 3) semantic annotation and 4) graph generation. The following paragraphs describe each of these steps in more detail.

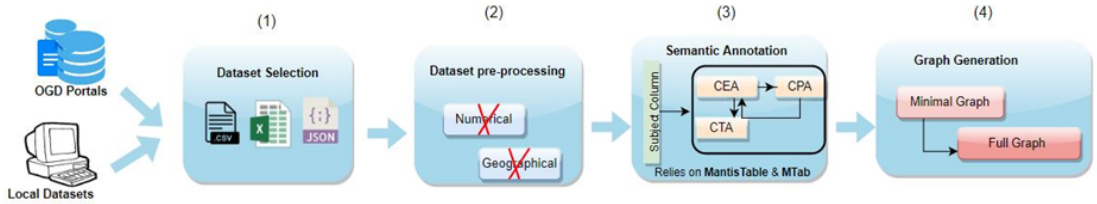


Figure 2: Overview of ODSAG prototype

During the **dataset selection step**, ODSAG takes as input an URL to access a dataset in Excel, CSV or JSON format on any open data portal (e.g., CKAN, OpenDataSoft, Socrata, DKAN) or on the user's computer (section (a) of Figure 3). Unlike SemanticBot [9], ODSAG allows users to interact with any open data portal (**R1**).

During the **data pre-processing step**, ODSAG removes all numeric columns (except columns containing postal code, NSI code, etc., which have a match in Wikidata), as well as geographic columns (**R2**), because currently most numeric and geographic values in datasets available on open data portals (e.g., the number of cases column in the COVID data) do not have a match in the used KG: Wikidata (section (b) of Figure 3).

The **semantic annotation step** aims at enabling users to automatically annotate an open dataset (section (c) of Figure 3). This step is divided into two sub-steps: the detection of the subject column (i.e., the column that is likely to have the most relations with the other columns) (**R3**) and the generation of CEA, CPA and CTA (**R4**). For the *subject column* detection, we use the procedure described in MantisTable [13]. This process starts by determining the literal columns (e.g., address, phone number, color, URL) using regular expressions. Once this step is complete, the system chooses from the remaining columns (called named entity columns) the subject column based on different statistic features, such as average number of words in each cell, fraction of empty cells in the column, fraction of cells with unique content and distance from the first named entity column [13]. More details on the subject column detection can be found in [13]. Once the subject column detection step is complete, we rely on Mtab to automatically generate the CEA, CPA and CTA. The mapping process in Mtab is done in three steps. Step 1 involves generating Wikidata resources using the Wikidata entity dump and history revisions, as well as creating two indexes for fuzzy entity search and fuzzy statement search. In Step 2, the fuzzy entity search is used to find relevant entity candidates for each cell in the table. Fuzzy statement search is used to handle the ambiguity of the table cells and consists of using the values of two cells in the same row to determine if there is a statement (relation) between them. In the end, only the entity candidates for which there is a relationship between the cells are retained. In Step 3, for each retained entity candidate, the system calculates a value matching (which depends on the statement similarities of each candidate and the other cells in that row) and keeps only the entity candidate with the highest value. Once all CEA are assigned, CPA are retrieved by aggregating all properties of statement candidates in the same rows, and then using majority voting to select the CPA annotations. For CTA annotations, we get the direct types from the CEA annotations in a column and vote for the majority types to get the CTA annotations. More details about each step of MTab can be found in [14]. Our

contribution in this step is that we have combined the strengths of MantisTable and MTab to perform both sub-steps. MTab does not offer a methodical subject column detection but has excellent results for semantic annotation and MantisTable does not offer excellent results for semantic annotation like MTab but has a consistent subject column detection.

During the **graph generation step**, ODSAG automatically creates two graphs (minimal graph and full graph) from the annotated data (R6). In the minimal graph, only the CTA and CPA are taken into account. CTA are represented as nodes and CPA as links connecting the subject column to other nodes (section (d) of Figure 3). This graph allows users to visualize the relations (CPA) between the subject column and other columns (CTA). The full graph, on the other hand, is an extended version of the minimal graph and includes all the information of the annotated data: CEA, CTA and CPA. CTA and CEA are represented as nodes and CPA as links connecting the subject column (resp. CEA of subject column) with other CTA (resp. CEA of other CTA) (section (e) of Figure 3). This graph allows users to visualize the relations between data content items and column names and to discover hidden relations between them. Both generated graphs are interactive, so users can move the nodes and discover relations between elements in a more readable way. The nodes in the graph are also clickable, allowing users to get more details about each entity if needed. Unlike some previous studies presented in Section 2 that generate only the minimal graph, ODSAG generates a full graph that helps users to better explore the relations between cell values (CEA).

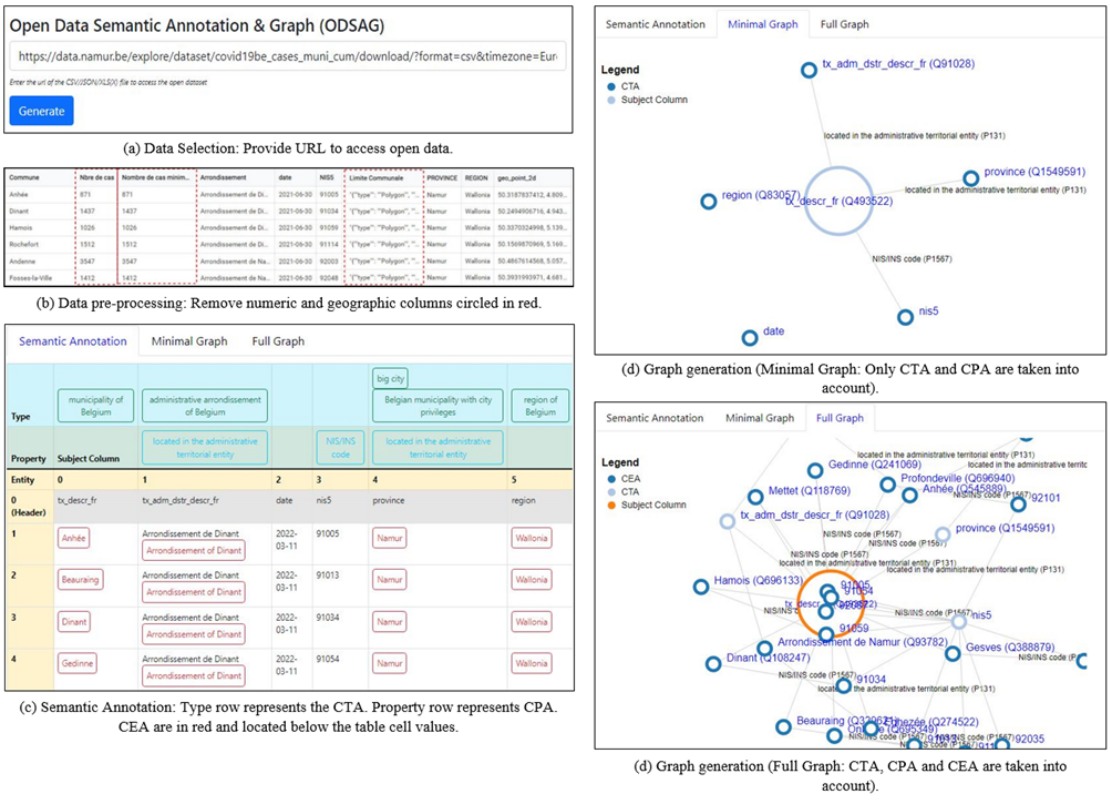
5. Implementation and Demonstration

In this section, we briefly explain the implementation of the ODSAG prototype (which source code is available on GitHub [3] and show a use case of the prototype.

Regarding the implementation, in order to provide a tool that is easy to install and use and that is interoperable with any open data portal (**R7**), we chose to implement: (1) a chrome extension that users can interact with (frontend) and (2) a django web application that is used to interact between the chrome extension and Mtab which can be hosted online or locally (backend).

Figure 3 illustrates an example of annotation and graphs generated by ODSAG when using the COVID1 open dataset available on the Namur (Belgium) portal. This portal was chosen as it is the most advanced portal in Wallonia (Belgium) and access with key stakeholders of this portal was possible (useful to evaluate later the direct integration of the tool to a portal). In order to annotate a dataset, the user must first load the extension in the chrome extensions (only for the first run). Then, as shown in section (a) of Figure 3, he/she has to go to any open data portal, copy the URL link of the desired open dataset and paste it into the ODSAG URL field, then click on “Generate” button. The system removes the numeric columns (“Nbre de cas”, “Nombre de cas minimum”) and geographic columns (“limite communale”, “geo_point_2d”) from the selected dataset (section (b) of Figure 3). A few seconds later, the system generates the annotated data and returns it as a table and graphs. Section (c) of Figure 3 shows the annotated data in table form where the **type** row includes the semantic tags associated with the name of each column (CTA). The **property** row includes the semantic tags associated with the relation between the subject

[3] https://github.com/chokkipaterne/odsag



(a) Data Selection: Provide URL to access open data.

(b) Data pre-processing: Remove numeric and geographic columns circled in red.

(c) Semantic Annotation: Type row represents the CTA. Property row represents CPA. CEA are in red and located below the table cell values.

(d) Graph generation (Minimal Graph: Only CTA and CPA are taken into account).

(d) Graph generation (Full Graph: CTA, CPA and CEA are taken into account).

Figure 3: Screenshots of ODSAG using a COVID Dataset from the Namur portal

column (tx_descr_fr) and the other columns. The **entity** annotations are in red and located below the table cell values. For example, the cell values "Anhée" and "Arrondissement de Dinant" have been respectively mapped to the entities Q545889 "Anhée" and Q93740 "Arrondissement of Dinant" (CEA), the column tx_descr_fr was associated with the class Q493522 "Municipality of Belgium" (CTA) and the system detects a property P131 "located in the administrative territory entity" between the column tx_descr_fr and tx_adm_dstr_descr_fr. Section (d) of Figure 3 shows the minimal graph. The subject column in this graph is represented by a light blue color, the CTA are represented by a dark blue color and the CPA are represented by the black links. Section (e) of Figure 3 shows the full graph. The subject column is represented by an orange color, the CTA are represented by a light blue color, the CEA are represented by a dark blue color and the CPA are represented by the black links.

6. Conclusion and Future Work

The aim of this paper was to facilitate the semantic annotation of open data in order to improve its discoverability and understanding. To achieve that goal, we first identified a list of 7 requirements that need to be implemented in a tool to facilitate semantic annotation of open data (see Table 2). Then, we implemented these requirements in a tool called ODSAG (Open Data Semantic

Annotation and Graph) and its effectiveness was evaluated with a COVID data from the Namur (Belgium) portal.

This research contributes to theory by proposing a list of requirements that need to be implemented in a tool to facilitate semantic annotation of open data (see Table 2). It also provides a comparative table highlighting the strengths and weaknesses of some existing semantic annotation tools used in the literature (see Table 1). This research also contributes to practice by implementing the identified requirements in a tool and providing the source code of the tool. This can be used as a starting point for developers to create their tool to facilitate semantic annotation of open data or to improve the prototype. However, this research has two main limitations that will need to be addressed in future work: the non-validation of the identified requirements and the non-evaluation of the proposed with the stakeholders (publishers and data analysts).

In the near future, we plan to validate the identified requirements, to test our prototype with other datasets and stakeholders, compare it with other existing tools, and extend the prototype with additional features, such as (1) integrating annotation of numerical and geographical columns, (2) integrating additional knowledge graphs such as DBPedia, LOV, Geonames and YAGO to improve the annotation and (3) generating a RDF file of the annotated open dataset for use in Linked Open Data.

Acknowledgments

The research was supported by a CERUNA PhD fellowship from the University of Namur.

References

[1] J. Attard, F. Orlandi, S. Scerri, S. Auer, A systematic review of open government data initiatives', Government Information Quarterly, Elsevier Ltd 32 (2015) 399–418.

[2] J. Bertot, P. Jaeger, J. Grimes, Using icts to create a culture of transparency: E-government and social media as openness and anti-corruption tools for societies', Government Information Quarterly, Elsevier B.V 27 (2010) 264–271.

[3] T. Davies, Open Data, Democracy and Public Sector Reform: A Look at Open Government Data Use from Data, Gov. Uk, 2010.

[4] P. Johnson, P. Robinson, Civic hackathons: Innovation, procurement, or civic engagement?', Review of Policy Research 31 (2014) 349–357.

[5] Results of semtab 2021', in: CEUR Workshop Proceedings, 2021, p. 1–12.

[6] P. Křemen, M. Nečaský, Improving discoverability of open government data with rich metadata descriptions using semantic government vocabulary', Journal of Web Semantics, Elsevier B.V 55 (2019) 1–20.

[7] A. Polleres, J. Umbrich, K. Figl, M. Beno, Perception of key barriers in using and publishing open data', JeDEM - EJournal of EDemocracy and Open Government 9 (2017) 134–165.

[8] M. Beno, E. Filtz, S. Kirrane, A. Polleres, Doc2rdfa: Semantic annotation for web documents', CEUR Workshop Proceedings 2451 (2019) 0–4.

[9] C. Bizer, T. Heath, T. Berners-Lee, Linked data - the story so far', International Journal on Semantic Web and Information Systems 5 (2009) 1–22.
[10] B. Moreau, N. Terpolilli, P. Serrano-alvarado, A semi-automatic tool for linked data integration', in: 18th International Semantic Web Conference (ISWC2019, 2019, p. 4–8.
[11] T. Knap, Towards odalic, a semantic table interpretation tool in the adequate project', in: CEUR Workshop Proceedings, 2017, p. 26–37.
[12] D. Roman, N. Nikolov, A. Putlier, D. Sukhobok, A. Berre, X. Ye, M. Dimitrov, Datagraft : One-stop-shop for open data management', Semantic Web Journal 9 (2016) 393–411.
[13] M. Cremaschi, R. Avogadro, A. Barazzetti, D. Chieregato, Mantistable se: An efficient approach for the semantic table interpretation', in: CEUR Workshop Proceedings, 2020.
[14] P. Nguyen, I. Yamada, N. Kertkeidkachorn, Semtab 2021 : Tabular data annotation with mtab tool', in: CEUR Workshop Proceedings, 2021.
[15] N. Abdelmageed, S. Schindler, Jentab meets semtab 2021's new challenges', in: CEUR Workshop Proceedings, 2021.
[16] Z. Zhang, Effective and efficient semantic table interpretation using tableminer', Semantic Web Journal 8 (2017) 921–957.