Process of generating RDF mapping model for OGD-LOD transformation

Khadidja Bouchelouche[1,*], Abdessamed Réda Ghomari[1] and Leila Zemmouchi-Ghomari[2]

[1]LMCS, Ecole nationale Supérieure d'Informatique (ESI), Algiers, Algeria.
[2]LTI, Ecole Nationale Supérieure des Technologies Avancées (ENSTA), Algiers, Algeria.

Abstract

The Transformation of Open Government Data (OGD) into Linked Open Data (LOD) can revolutionize how we access and use OGD since the LOD technology guides the publication of data and its interconnection in a machine-readable medium, allowing automatic interpretation and exploitation. Considering the nature of OGD, which is often unstructured, heterogeneous, and significant in volume, this requires an effort of integration to transform OGD into LOD. Among the essential issues that must be addressed is the generation of the Resource Description Framework (RDF) mapping model, which requires extracting RDF triples from OGD. This is a tricky process in OGD-LOD transformation due to the variety of OGD formats. Thus, many works addressed the case of Relational database RDF mapping, considering its helpful structure for extracting RDF triples. For this, it is necessary to provide a model that can generate RDF mapping for different input data formats to extract the RDF triples and link the input datasets to other datasets on the web.

This paper proposes a new approach for generating an RDF mapping model for OGD-LOD transformation. It can be used to transform any OGD data into LOD, regardless of its format (CSV, Excel, HTML, PDF, and TXT), using the interlinking techniques such as Named Entity Recognition (NER) and DBpedia alignment and including the four principles of linked data as a validation layer. We believe that it has the potential to significantly accelerate the adoption of LOD and make it more accessible to a broader range of users.

Keywords

Open Government Data (OGD), Linked Open Data (LOD), OGD-LOD transformation, RDF mapping model

1. Introduction

Open Government Data (OGD) is an international collaboration between the United States, United Kingdom, France, and Singapore governments to share machine-readable datasets covering government activities [1]. The datasets are produced by governments or under the control of government entities [2].

Many datasets could belong to government data, including data held indirectly by public administrations (e.g., through agencies or subsidiaries), such as pollution/climate, education/childcare, and traffic/congestion [3], [4].

A large number of applications have been developed that exploit the OGD (https://www.data.gov/applications) in different countries and offer many services to people wishing to obtain practical information concerning, for example, the distribution of job applications by sector of activity and by region of a country or electricity consumption according to the type of household appliance used by time slot of the day in another country or even more generally the foods to avoid or to advocate in the case of this or that disease.

OGD is often unstructured, heterogeneous, and significant in volume. This requires an effort of integration to Transform OGD into Linked Open Data (LOD). LOD is derived from combining open-linked data [5]. LOD is based on realizing the large-scale implementation of a lightweight Semantic Web [5].

TACC'2024, The 4th Tunisian-Algerian Conference on applied Computing, December 17-18, 2024, Constantine, Algeria
*Corresponding author.
k_bouchelouche@esi.dz (K. Bouchelouche); a_ghomari@esi.dz (A. R. Ghomari); leila.ghomari@ensta.edu.dz (L. Zemmouchi-Ghomari)
0000-0002-8962-9838 (K. Bouchelouche); 0000-0002-0683-9304 (A. R. Ghomari); 0000-0002-6754-6062 (L. Zemmouchi-Ghomari)

The Linked Data (LD) initiative provides a framework in which data are represented, connected, and automatically accessed and processed by applications or web services. Thus, the linked data principles allow data publication in a self-descriptive mean and facilitate the integration of data from different sources [6]. In addition, linked data facilitates data discovery and consumption and reduces redundancy [6].

To transform OGD into LOD, the process of generating an RDF (Resource Description Framework) mapping model is required to extract the RDF triples from the input data files [7], [8].

Many works were conducted to provide an RDF mapping language such as [9], [10], [7], [11], [12] and [13], where the mapping is done manually and to specific input data formats [14], [15], [7], [12], [10].

This process is considered a difficult stage in the Process of the OGD-LOD transformation since the input data formats are various and of several data types [9], [16], [17].

It is challenging to provide a model that can generate RDF mapping for mixed data formats, to extract RDF triples and link them to external datasets on the web, without requiring intensive human workload [12], [9], [16], [17], [18], [19].

In this paper, we aim to cover this gap by providing an RDF mapping model to extract RDF triples and link them to external datasets on the web, for structured and unstructured data formats and types, as well as reducing human intervention in the mapping task.

The organization of this paper is as follows: Section II presents the Related Works. Next, section III presents the proposed RDF mapping process. Finally, the conclusion summarizes the work and the findings in section IV.

2. Related Works

This section presents the current approaches to RDF mapping. The works in this field tended to provide RDF mapping languages for Relational databases.

Thus, many works were conducted to provide RDF mapping languages such as [7], [9], [12] and [10]. Where the RDF mapping was done manually and to specific input data formats such as Relational Databases [15], [13], [11], [14], [7], [12], [10].

In [15], the authors provided a survey comparing the approaches allowing RDF mapping from Relational databases, based on a defined reference framework comprising: mapping generation, query execution, and data integration achieved by mapping Relational databases to RDF.

In [13], the authors provided a comparison framework based on use cases and requirements for mapping Relational Databases to RDF languages, where nine RDF mapping languages were treated: Direct Mapping, eD2R, R2O, Relational.OWL, Virtuoso RDF Views, D2RQ, Triplify, R2RML, R3M. As a result, the authors provided a classification of RDF mapping languages from Relational databases into four categories: direct mapping, read-only general-purpose mapping, read-write general-purpose mapping, and special-purpose mapping.

In [11],the authors provided a survey comparing the approaches to allowing RDF mapping from Relational databases based on a defined reference framework comprising mapping generation, query execution, and data integration achieved by mapping Relational databases to RDF.

In [14], the authors presented a series of reusable RDF mapping patterns from Relational databases, based on the author's experience, where the mappings were represented in the R2RML language.

In [7], the authors described the xR2RML, a language for expressing RDF mapping from various types of databases (XML, Object-Oriented, NoSQL).
xR2RML is an extension of the R2RML mapping language, which relies on the properties of the RML mapping language. R2RML treats the RDF mapping based on relational databases. In contrast, RML extends R2RML to treat the RDF mapping of heterogeneous data formats (CSV, XML, JSON) that do not include the case of dealing with different types of heterogeneous databases. Thus, xR2RML extends this scope to a wide range of non-Relational databases.

In [12], the authors presented YAMA Mapping Language (YAMAML) as a lightweight RDF mapping language, which is based on Yet Another Metadata Application Profiles (YAMA). YAMA is an extensible intermediary application that generates application profile expressions (a combination of vocabularies, mixed and matched based on different name spaces and optimized for a particular local application). YAMAML allows the map of RDF data structures to RDF based on the application profile. It is proposed as an intermediary format for generating RDF but does not consider RDF representation syntax in the output.

In [10], the authors proposed using the RDF Mapping Language (RML) to transform Dublin Core descriptions of (X)HTML web pages into an RDF model.

From the presented approaches of RDF mapping, we can notice the need for intensive human intervention for the mapping tasks or the consideration of particular data formats (Relational databases, different types of databases, or Dublin Core descriptions of (X)HTML). The necessity of providing an RDF mapping model comes to address the issue of the different input data formats (structured and unstructured) to extract the RDF triples and link the input datasets to other datasets on the web by reducing human intervention through a process to automate this task. Considering this task of RDF mapping requires an intensive human workload to consider the possible input formats of the data as well as the type of the data itself [12], [9], [16], [17].

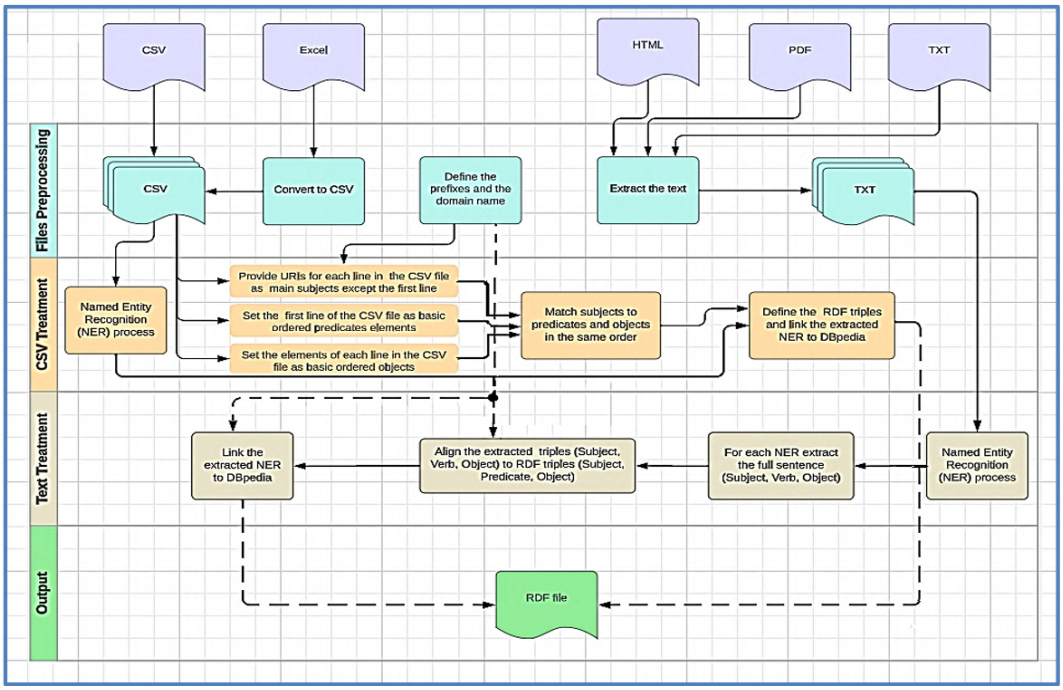

Figure 1: The process of generating RDF mapping.

3. The proposed RDF mapping process

3.1. Overview and Objectives

This section proposes an RDF mapping process to improve the OGD-LOD transformation without requiring intensive human workloads.

The necessary Terminology and vocabulary for this process are defined as follows:

- Named Entity Recognition (NER): helps identify predefined entities in a text and is a fundamental step in many tasks, like building knowledge graphs or answering questions [20].
- Subject-Predicate-Object alignment: This allows the generation of the Subject-Predicate-Object from a text based on the extracted NER and associated information in the text.

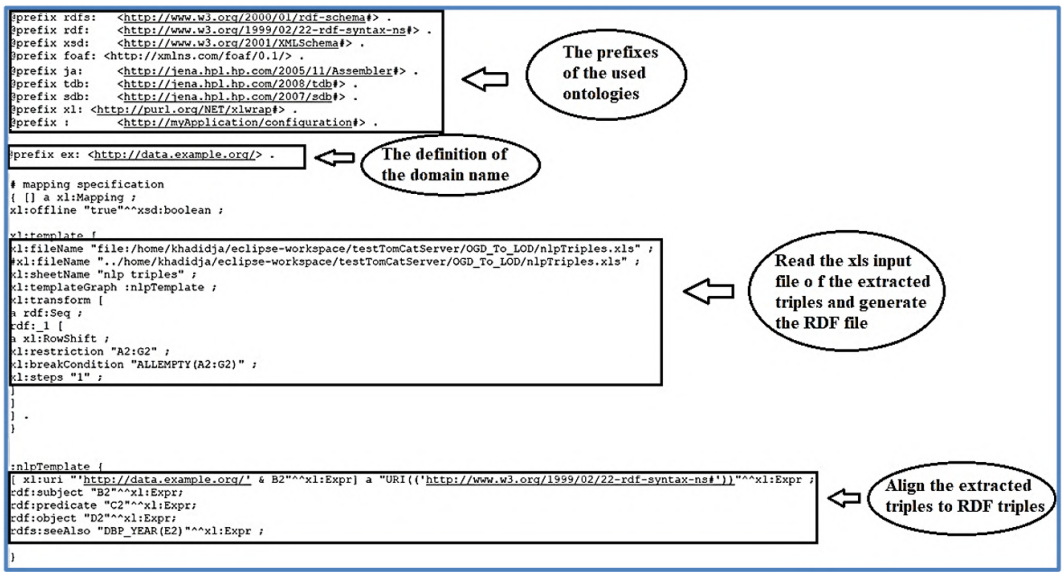

Figure 2: The proposed mapping model.

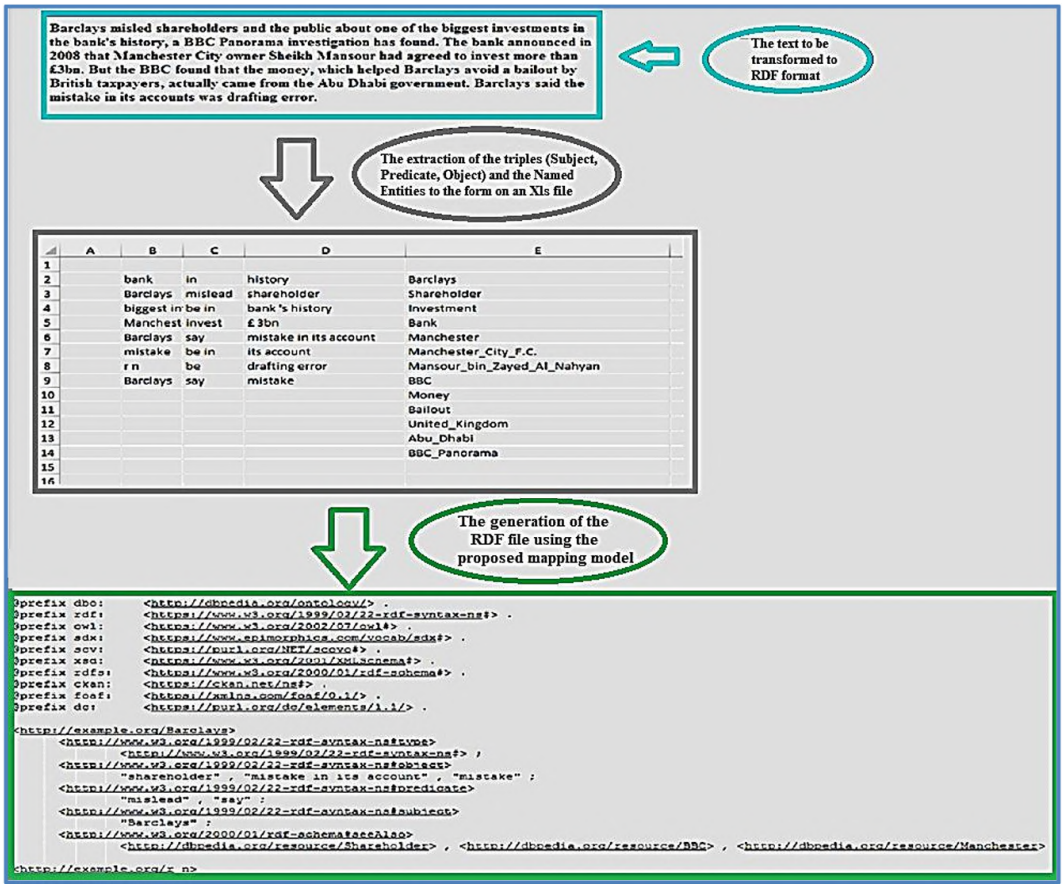

Figure 3: An example of transforming text to RDF using the proposed mapping model.

- RDF Mapping: This represents the template that enables the generation of RDF triples and associated links from well-structured formats like CSV.
- OGD-LOD Transformation: is the process of converting OGD into LOD (RDF format).

The proposed process considers three main objectives:

- The treatment of heterogeneous OGD formats (CSV, Excel, HTML, PDF and TXT).

- The generation of a single RDF file, linked to other datasets on the web based on DBPedia database.
- The validation of the generated RDF file according to the linked data principles.

3.2. RDF Mapping Process

To provide a durable model that allows the mapping of OGD into LOD based on RDF graphs, we define the following rules:

- The input data files must be in English.
- The RDF mapping process must consider all the input formats for a generic process to reduce human intervention.
- The prefixes of the used ontologies must be defined in advance and general to consider the different types of data.
- The domain name must be defined.

Figure 1 represents the process for generating RDF mapping for the OGD-LOD transformation.

Generating RDF mapping (See Figure 1), begins with the different input files (CSV, Excel, HTML, PDF, and TXT) of the Purple color. Then, the input files will be treated according to the preprocessing stage, which is composed of the following treatments:

1. For the structured data (CSV and Excel), convert the Excel formats to CSV formats, to consider a unified CSV formats. Many libraries are provided for this conversion such as Pandas[1], CSV file API[2], xlrd[3] in python, and Aspoce[4] in java.
2. For the unstructured data (HTML, PDF and TXT), extract the text from the HTML and PDF formats to treat them as a unified TXT formats, using IRONPDF[5] library in java and PyPDF2[6], Pdfminer[7], Pdfplumber[8] libraries in python.
3. Define the prefixes and the domain name to use for the mapping and linking process of the different data types such as: dbo[9], rdf[10], owl[11], sdx[12], scv[13], xsd[14], rdfs[15], ckan[16], foaf[17], dc[18].

After that, the set of CSV files will be treated according to their formats (Orange color in Figure 1), as follows:

1. Use NER to identify named entities in CSV files, such as names of persons, countries, organizations, and institutions, with libraries like SpaCy[19], NLTK[20] and Flair[21] (Python) or OpenNLP[22] and Stanford NER[23] (Java).

[1]https://pandas.pydata.org/
[2]https://docs.python.org/3/library/csv.html
[3]https://xlrd.readthedocs.io/en/latest/
[4]https://products.aspose.com/cells/java/
[5]https://ironpdf.com/java/examples/extract-text-from-pdf/
[6]https://pypi.org/project/PyPDF2/
[7]https://pypi.org/project/pdfminer/
[8]https://pypi.org/project/pdfplumber/
[9]<http://dbpedia.org/ontology/>
[10]<https://www.w3.org/1999/02/22-rdf-syntax-ns>
[11]<https://www.w3.org/2002/07/owl>
[12]<https://www.epimorphics.com/vocab/sdx>
[13]<https://purl.org/NET/scovo>
[14]<https://www.w3.org/2001/XMLSchema>
[15]<https://www.w3.org/2000/01/rdf-schema>
[16]<https://ckan.net/ns>
[17]<https://xmlns.com/foaf/0.1/>
[18]<https://purl.org/dc/elements/1.1/>
[19]https://realpython.com/natural-language-processing-spacy-python/
[20]https://www.nltk.org/
[21]https://flairnlp.github.io/
[22]https://opennlp.sourceforge.net/projects.html
[23]https://nlp.stanford.edu/software/CRF-NER.shtml

2. Provide URIs for each line in the CSV file as main subjects, except the first line.
3. Set the elements of the first line in the CSV file as basic ordered predicates.
4. Set the elements of each line in the CSV file(Except the first line), as basic ordered objects.
5. Match the subjects to predicates and objects in the same order using the proposed mapping model to generate the RDF triples (Figure 2).
6. link the extracted NER to DBpedia database and generate the RDF file.

For the TXT files, they will be treated according to their formats (Grey color in Figure 1), as follows:

1. Use NER to identify named entities in TXT files, such as names of persons, countries, organizations, and institutions, with libraries like SpaCy[24], NLTK[20] and Flair[21] (Python) or OpenNLP[22] and Stanford NER[1] (Java).
2. For each extracted NER, extract the full sentences (Subject, Verb, Object) using OpenIE[25] API for Java.
3. Align the extracted sentences (Subject, Verb, Object) to RDF triples (Subject, Predicate, Object) using the proposed mapping model (Figure 2).
4. Link the extracted NER to DBpedia database.

The output for both file formats (CSV and TXT) will be an RDF file (Green color in Figure 1).

The proposed mapping model presented in Figure 2, considers the case of transforming several formats (CSV, Excel, Txt, PDF and HTML) to RDF using the XLWrap[26] tool since it is efficient for the transformation of spreadsheets to arbitrary RDF graphs based on a mapping specification, as it supports Microsoft Excel and OpenDocument spreadsheets such as CSV files. Moreover, it can load local files or download remote files via HTTP.

Figure 3 presents an example of transforming a text automatically to RDF based on the proposed mapping model.

3.3. Validation Process

We can evaluate the efficiency of the proposed process based on the validity of its generated RDF file (linked data format) of Figure 3.

Thus,the generated RDF file is evaluated and validated according to the four principles using the Ontology-Evaluation/LD-Principles[27] , which was proposed in [6] as follows (see Figures 4, 5, 6 and 7):

- Principle 1: extracts all triples from the RDF file. Then check their representation with valid URIs. The validation is applied by replacing all detected no URIs with URIs that identify them.
- Principle 2: extracts RDF triples that comply with the first principle and are dereferenceable HTTP URIs (HTTP code testing with an agent to get response).
- Principle 3: checks the queried resource for providing valuable information and validate the RDF syntax of the resource.
- Principle 4: verifies that the datasets include links to external datasets. DBpedia is queried via its endpoint for equivalent URIs.

According to the first principle, the evaluation of the generated RDF file has been validated with a 92.7% score, as shown in Figure 4.

While the evaluation of the latter according to the second principle has been validated with a 100% score, as shown in Figure 5.

For the third principle, the evaluation has been validated by detecting the existence of 45 examples of helpful information (see Figure 6).

[24]https://realpython.com/natural-language-processing-spacy-python/
[25]https://stanfordnlp.github.io/CoreNLP/openie.html
[26]https://xlwrap.sourceforge.io/?fbclid=IwAR2RwHnPrpdZLdsBc1hByAqXwhleR59gzReWXsOvnjVdHc6tNnVuEmhGcexample
[27]https://sourceforge.net/projects/evaluate-ontology-ldprinciples/

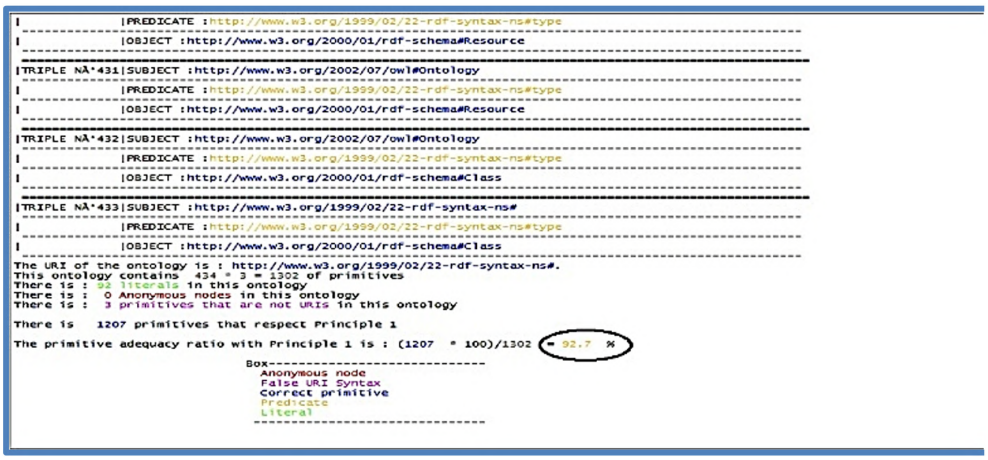

Figure 4: The evaluation of the RDF file according to the first principle.

Figure 5: The evaluation of the RDF file according to the second principle.

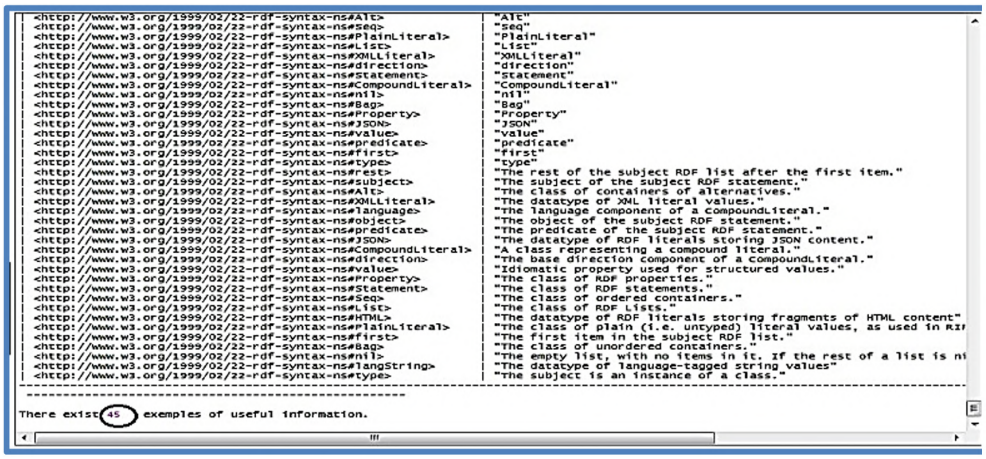

Figure 6: The evaluation of the RDF file according to the third principle.

For validating the generated RDF file according to the fourth principle, the evaluation has been validated with a 7.37% score (see Figure 7).
While the average rate of the generated RDF file according to the four principles has been validated with a 61.27% score, as shown in Figure 7.

Figure 7: The evaluation of the RDF file according to the fourth principle.

Table 1
Comparing the proposed RDF mapping approach to the existing approaches.

RDF mapping approaches	Variety of input formats	Human intervention level	RDF mapping reusability
[11]	Relational database format	High	None
[14]	Relational database format	High	Yes
[7]	XML, Object-Oriented, NoSQL database formats	High	None
[12]	Non-RDF data structures	Medium	None
[10]	Dublin Core descriptions of (X)HTML	Medium	None
The proposed RDF mapping approach	CSV, Excel, HTML, PDF, and TXT formats	Low	Yes

3.4. Comparison with Previous Approaches

To show what our approach brings as novelty compared to previous approaches, Table 1 presents a comparison of the proposed approach compared to previous work by following these criteria:

- The variety of input formats: to check the input data formats addressed for the RDF mapping generation.
- The level of Human intervention for the generation of RDF mapping: we consider three main levels, ”High” which requires to adjust the RDF model before running the process, ”Medium” which requires adjusting some parameters via the interface before execution and ”Low” which must be generated once and then be reused without needing for adjustment before execution.
- The possibility of reusing the RDF mapping model: to evaluate whether the generated RDF mapping model is not specific to one dataset, but can be reused directly on other datasets in order to automate this process.

From Table 1, we can notice the novelty of the proposed RDF mapping approach according to the three main criteria.

The proposed RDF mapping approach considers heterogeneous data formats for structured and unstructured data. It reduces the Human intervention for the mapping generation since it provides a general process that can address the issue of mapping RDF data as a single model. Thus, the proposed RDF mapping approach also enables the re-usability of the RDF mapping model since it is defined as a general approach that does not depend on the data types or structures.

4. Conclusion

This paper aims to improve the OGD mapping in heterogeneous formats into RDF formats. Thus, we provided a generic process for RDF mapping that reduces human intervention and facilitates the extraction of RDF triples.

We believe this proposition will be helpful for developers. Indeed, we presented libraries for implementing the Natural Language Processing (NLP) and NER tasks in the process for Java and Python programming languages.

Nevertheless, other tasks must be implemented from scratch, such as matching the extracted entities to RDF triples and linking the Entities to the DBpedia database.

In future work, we aim to consider converting visual elements (tables, charts) from a PDF to text. To improve RDF data quality, we plan to apply advanced preprocessing techniques to analyze document structure, extracting metadata and title hierarchies before RDF mapping. We will also test OpenIE Performance for extracting relationships in unstructured text. AI techniques, especially Transformers and pre-trained language models like BERT, may achieve improved results.

Acknowledgements

This work has been carried out within the framework of the PRFU Project OGDIVAA[28] (Open Government Data Initiatives and Value delivery for Algerian public Agencies).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] C. V. Buttow, S. Weerts, Open government data: The oecd's swiss army knife in the transformation of government, Policy Internet 14 (2022) 219–234.

[2] J. Attard, F. Orlandi, S. Auer, Data driven governments: Creating value through open government data, in: Transactions On Large-Scale Data-and Knowledge-Centered Systems XXVII: Special Issue On Big Data For Complex Urban Systems, 2016, pp. 84–110.

[3] K. Bouchelouche, A. R. Ghomari, L. Zemmouchi-Ghomari, Enhanced analysis of open government data: Proposed metrics for improving data quality assessment, in: 2022 5th International Symposium On Informatics And Its Applications (ISIA), 2022, pp. 1–6.

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

[5] R. Nawi, S. Noah, L. Zakaria, Integration of linked open data in collaborative group recommender systems, IEEE Access 9 (2021) 150753–150767.

[6] L. Zemmouchi-Ghomari, K. Mezaache, M. Oumessad, Ontology assessment based on linked data principles, International Journal Of Web Information Systems 14 (2018) 453–479.

[7] F. Michel, L. Djimenou, C. Zucker, J. Montagnat, xr2rml: Relational and non-relational databases to rdf mapping language, 2017.

[8] P. Heyvaert, D. Chaves-Fraga, F. Priyatna, O. Corcho, E. Mannens, R. Verborgh, A. Dimou, Conformance test cases for the rdf mapping language (rml), in: Iberoamerican Knowledge Graphs And Semantic Web Conference, 2019, pp. 162–173.

[9] A. Dimou, M. V. Sande, P. Colpaert, R. Verborgh, E. Mannens, R. Walle, Rdf mapping language (rml), W3C, Unofficial Draft, 2020.

[28] https://sites.google.com/esi.dz/ogdivaa/accueil

[10] T. Georgieva-Trifonova, Transforming dublin core (x) html descriptions to rdf model using rdf mapping language, in: 2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH), 2023, pp. 1–5.
[11] M. Arenas, A. Bertails, E. Prud'hommeaux, J. Sequeda, Others, A direct mapping of relational data to rdf, W3C Recommendation, 2012.
[12] N. Thalhath, M. Nagamori, T. Sakaguchi, Yamaml: An application profile based lightweight rdf mapping language, in: International Conference On Asian Digital Libraries, 2022, pp. 412–420.
[13] M. Hert, G. Reif, H. Gall, A comparison of rdb-to-rdf mapping languages, in: Proceedings Of The 7th International Conference On Semantic Systems, 2011, pp. 25–32.
[14] J. Sequeda, F. Priyatna, B. Villazón-Terrazas, Relational database to rdf mapping patterns, in: WOP, 2012.
[15] S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. T. Jr, S. Auer, J. Sequeda, A. Ezzat, A survey of current approaches for mapping of relational databases to rdf, W3C RDB2RDF Incubator Group Report, 2009.
[16] M. Vafopoulos, S. Rallis, I. Anagnostopoulos, V. Peristeras, D. Negkas, I. Skaros, A. Tzani, Mining and linking open economic data from governmental communities, in: Open Source Systems: Enterprise Software And Solutions: 14th IFIP WG 2.13 International Conference, OSS 2018, 2018, pp. 144–148.
[17] P. Budsapawanich, C. Anutariya, C. Haruechaiyasak, A conceptual framework for linking open government data based-on geolocation: A case of thailand, in: Semantic Technology: 8th Joint International Conference, JIST 2018, 2018, pp. 352–366.
[18] K. Bouchelouche, A. R. Ghomari, L. Zemmouchi-Ghomari, Open government data (ogd) publication as linked open data (lod): A survey, International Journal Of Computer And Information Technology 10 (2021).
[19] I. Mutambik, A. Almuqrin, J. Lee, J. Gauthier, A. Homadi, Open government data in gulf cooperation council countries: An analysis of progress, Sustainability 14 (2022) 7200.
[20] V. Moscato, M. Postiglione, G. Sperlí, Few-shot named entity recognition: Definition, taxonomy and research directions, ACM Transactions on Intelligent Systems and Technology 14 (2023).