Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230014

Classification of Linking Problem Types for Linking Semantic Data

Raphaël CONDE SALAZAR [a,1], Clément JONQUET [a,b] and Danai SYMEONIDOU [a]
[a] *MISTEA, University of Montpellier, INRAE & Institut Agro, France*
[b] *LIRMM, University of Montpellier & CNRS, France*
ORCiD ID: Raphaël Conde Salazar https://orcid.org/0000-0002-6926-5299, Clément Jonquet https://orcid.org/0000-0002-2404-1582, Danai Symeonidou https://orcid.org/0000-0003-1152-5200

Abstract. As the number of RDF datasets published on the semantic web continues to grow, it becomes increasingly important to efficiently link similar entities between these datasets. However, the performance of existing data linking tools, often developed for general purposes, seems to have reached a plateau, suggesting the need for more modular and efficient solutions. In this paper, we propose –and formalize in OWL– a classification of the different Linking Problem Types (LPTs) to help the linked data community identify upstream the problems and develop more efficient solutions. Our classification is based on the description of heterogeneity reported in the literature –especially five articles– and identifies five main types of linking problems: predicate value problems, predicate problems, class problems, subgraph problems, and graph problems. By classifying LPTs, we provide a framework for understanding and addressing the challenges associated with semantic data linking. It can be used to develop new solutions based on existing modularized tools addressing specific LPTs, thus improving the overall efficiency of data linking.

Keywords. Data linking, Semantic web, Linking Problem Types, Classification

1. Introduction

For more than twenty years, important work has been going on for the development of the semantic web [1] with the aim of sharing data online and facilitating access by machines to human knowledge. In this approach, Linked Open Data (LOD) promotes the sharing and reuse of royalty-free datasets, based on the semantic web model and tools, such as the RDF and OWL representation languages. But while the number of datasets available as LOD is increasing every year, a new challenge must be met: data linking. Indeed, in order to maximize the knowledge from a resource, agents browsing these datasets must be able to link two resources designating the same thing but identified by distinct identifiers (URIs) within each of datasets. For example, a prominent actor might venture into politics, resulting in their inclusion and description within separate knowledge bases for cinema and politics. In order to write his biography, the two URIs, generally distinct, which identify this same person must be linked-back by an equivalence link such as the semantic relation `owl:sameAs` whose uses are described in [2,3,4,5].

[1] Corresponding Author: Raphaël Conde Salazar, raphael.condesalazar@online.fr

The very fact that data linking is based on *similarity* is problematic if one considers that it is a subjective notion that is difficult to grasp in a formal way. Indeed, similarity does not express exact likeness but a close resemblance or similitude for which automated processing must then be parameterized in an equally subjective manner. For example, let us take the descriptions in two datasets of two distinct persons homonyms by their first and last names. Should we consider semantically that they are the same person if their date of birth is identical or should we consider the low probability that these two homonyms were born on the same day. In addition to the difficulty of setting up decision trees, the *heterogeneity* of the datasets is also a major obstacle to data linking. Indeed, taking the previous example, one would like to choose one or more characteristics that would uniquely identify the two persons such as their social security number, but would be embarrassed if this property were described using two similar, yet formally different, predicates (e.g., `hasSocialSecurityNumber` and `hasSSN`) and/or if the value of this property was presented in a different format (e.g., the number 1880475114782 and the literal "1-88-04-75-114-782"). The search for similarity between two (ontological) entities is therefore strongly impacted by the different semantic, lexical, or structural heterogeneities that can be obtained from the design of the datasets given the constraints imposed by RDF. The non-respect of good practices such as the non-use of language tags for labels, or serialization errors such as the presence of duplicate identification keys, can also reinforce these heterogeneity problems. All these heterogeneities make data linking based on similarity more complex and tedious, and also require the intervention of experts.

Several data linking tools have been developed according to different strategies [6], and are confronted during benchmarking campaigns such as the Ontology Alignment Evaluation Initiative (OAEI) [7], an annual event to evaluate ontology alignment and data linking tools. In general, the competing tools offer very generic solutions composed of several modules in an attempt to resolve a maximum number of types of heterogeneity presented in these tests. Although high, the maximum efficiency of these tools seems to have stabilized in recent years without reaching a fully reliable ideal solution as pointed out by Algergawy et al. [8] and Pour et al. [9,10] in the conclusions of their presentations of the OAEI benchmark results for the years 2019, 2020 and 2021.

In this paper, we suggest and anticipate a novel technique for data linking, which involves creating profiles for pairs of datasets and utilizing machine learning algorithms to recommend which modular solutions would be best suited to these profiles for the linking task. This approach deviates from the conventional incremental methods currently used and takes advantage of already existing data linking tools and datasets. To establish these profiles, all the problems that can be encountered when linking two data sets must be identified. We therefore propose a classification of the different types of Linking Problem Types (LPTs) that can be encountered during semantic data linking. This classification of LPTs, also formalized in OWL, will be publicly accessible to the community for inclusion in automatic tools and future improvements. To the best of our knowledge, there is no such a formalized classification of the types of semantic data binding or linking problems and we believe that this is an impediment to a fully automated treatment of data linking, especially with new machine-learning based approach coming.

The rest of the article is organized as follows: we present related work in Section 2. Then, we present our vision of similarity when linking two RDF entities in Section 3, as well as the methodology used to build our classification from the different types of

heterogeneities coming from the data linking corpus in Section 4. Section 5, details our resulting classification and its formalization in OWL. Finally, we discuss perspectives and conclude our work in Section 6.

2. Related work

Data linking has been defined for example as:“ the task of establishing typed links between entities across different RDF datasets via the help of automatic link discovery systems .” [11]. In [12], the authors distinguish two approaches to data linking: “(i) A similarity-based approach in which the more similar two resources are, the more likely they are to be linked; (ii) A key-based approach in which a key determines the identity of a resource: two resources with the same key must be linked”. In this paper, we define data linking as the task of establishing similarity or hierarchical relationships between distinctly identified entities in two different semantic datasets.
Here is a summary of some common data linking techniques:

- *Deterministic linking* involves linking dataset records based on unique identifiers (such as the social security number in the previous example) or other unique identifiers that allow for a one-to-one match between entities in different datasets. This method is considered the most accurate and efficient but requires the use of common unique identifiers across datasets which may not exists or may be hard to identify.
- *Probabilistic linking* involves matching entities based on non-unique identifier properties such as names, addresses, and dates of birth. The technique calculates the probability that two records refer to the same entity.However, determining a unique key on the basis of several pairs of properties and values remains an arduous task.Work has been carried out for the automated determination of these keys [13,14,15]. Probabilistic linking is useful when unique identifiers are not available, but it is less accurate than deterministic linking.
- *Rule-based linking* involves defining rules that specify the conditions for linking records between datasets [16,17]. For example: two records match if they have the same name and address. Rule-based linking can be useful when there is a high degree of certainty about the conditions for linking records.
- *Knowledge graph embedding* involves representing relationships as translations in the embedding space [18]. Graphs are transposed into vector spaces because the latter offer a wider range of tools for mathematical and statistical processing. This means that technologies such as machine learning can be applied more easily to these graphs. This is one of the most recent techniques in the field of data liking, but despite the many advantages it has over other, more traditional techniques, it still has limitations, such as the fact that this method does not hold up well when the relational paths are long or complex [19].

3. What it is for two RDF resources to be similar?

According to the Larousse French dictionary, similar things are defined as follows: “A set of things that can, in a certain way, be assimilated to each other”. In the following, we review how this definition may be applied to RDF resources belonging to different datasets by trying to clarify this notion “in a certain way”.

Harispe et al. [20] say:“Similarity assessment must therefore not be understood as an attempt to compare object realisations through the evaluation of their properties, but rather as a process aiming to compare objects as they are understood by the agent which

estimates the similarity (e.g., a person, an algorithm). The notion of similarity therefore only makes sense according to the consideration of a partial (mental) representation on which the estimation of object similarity is based". We agree with this last quote and the fact that the simple observation of common characteristics between two entities is not enough to make them similar. Indeed, if we are interested in two person whose descriptions indicate that their height is precisely 1m78 this characteristic would not be sufficient to allow us to affirm that they are the same person. Nevertheless, the observation of an entity's characteristics is primordial in a similarity search, because one or a set of characteristics can make an entity unique through its description (e.g., social security number, or the set of first name, last name, date and place of birth, eye color and postal address). The observation of certain characteristics can also invalidate our search for similarity (e.g., we will not try to compare two person if their description reports that one has blue eyes and the other brown). This use of an entity's characteristics in a similarity search implies de facto that we can compare the same characteristic and its value in a similar way. For our objective of setting up a classification of LPTs, we are thus simplistically focused on the comparison of one and the same characteristic of two RDF resources supposed to represent the same entity through its description in two distinct RDF datasets which can also not be obvious as we will see.

Let two RDF triples (see Fig. 1) then, S and S' are considered similar if they share a common characteristic, which implies that P and P' are similar as well as O and O'.

Figure 1. Two RDF triples.

We intuitively identified four types of issues for similarity between S and S':

1. If P=P', then the issue is in establishing similarity between O and O' (see Fig. 2.a). The problem can be linguistic or structural when the object is a literal. For examples, the value of the property `hasForQualification` of a person can be described via the literal "coach operator" in one dataset and "bus driver" in another. Or the value of the property `dateOfBirth` of a person can be described via the literal "24 march 2023" in one dataset and "2023-03-24" in another due to a difference in the date format.

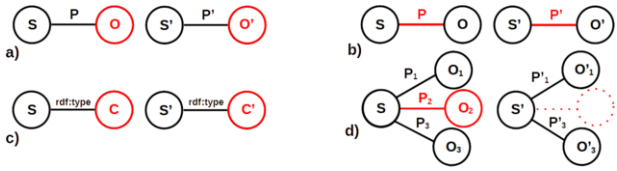

Figure 2. **a)** First intuitive issue: Objects are different while subjects and predicates are the same.
b) Second intuitive issue: Predicates are different while subjects and objects are the same.
c) Third intuitive issue: The subjects are identical but belong to different classes.
d) Fourth intuitive issue: Missing characteristics from the description of one of the subjects.

2. If O=O', then the issue is in establishing similarity between P and P' (see Fig. 2.b). The problem can be linguistic, structural or semantic. At the linguistic level, for example, the professional qualification of a person could be expressed through the

predicate `hasForQualification` in a dataset mostly in English whereas the equivalent predicate may be `aPourQualification` in another dataset mostly in French. At the structural level, for example, a characteristic of a person like his date of birth can be represented by the property `dateOfBirth` in a dataset and this same characteristic by the three properties `monthOfBirth`, `dayOfBirth` and `yearOfBirth` in another dataset. At the semantic level, for example, the name of a person can be described by the relation `foaf:name` from the FOAF vocabulary[2] in a dataset and by the relation `vcard:fn` of the vCard ontology[3] in another one.

3. If predicates indicates that the subject is the instance of a class (e.g., `rdf:type`), then the issue is in establishing similarity between the types of an entity C and C' (see Fig. 2.c). The problem may be one of terminology or specialisation/generalisation. For example, a person can belong to two different subclasses of a given class, it can be an instance of the class `Person` and an instance of the class `Actor`, both subclasses of the class `Human`.
4. Issue in establishing similarity of entities when a property of S is absent for S' (see Fig. 2.d). For example, a person can be described with his name, date of birth and social security number in one dataset and only with his name and date of birth in another.

We find these four issues originate in the diversity of values, structure, and logic used in the development of the compared datasets. These types of problems are known and arise from the flexibility of RDF which, as we have seen in the previous examples, does not impose any constraints on the data, just formalize how to encode them. These types of terminological and structural problems have been reported in the context of XML exploitation and are described in the literature under the term heterogeneity. The RDF syntax (i.e., RDF/XML) is based on XML and therefore inherits these heterogeneity problems. We will then use these four issues to initialize our classification of LPTs.

4. Methodology used for the construction of the LPTs classification

We will now confront the four issues previously discussed with the different types of heterogeneities reported in the data linking literature. To achieve this objective, we start by building a small corpus of articles about data linking, then from this corpus, we keep only the articles dealing with the heterogeneities that can be encountered during semantic data linking.

4.1. Analysis of data linking literature

We did a systematic review of data linking literature in order to compile an exhaustive list of LPTs. Figure 3 illustrates our approach: we started from a very specific term (e.g., "OAEI") with which we performed a first bibliographic search with Google Scholar and Web of Science to create a corpus of research papers (as PDF documents). Then the extracted corpus is fed to a text mining tool called Gargantext[4], to obtain a list of words that are considered statistically relevant to the topic covered by this corpus. See, for example, Table 1 for the results obtained for the expression "OAEI". This list of all terms is then re-injected into a bibliographic search whose articles obtained are again re-injected

[2] Friend Of A Friend vocabulary. http://www.foaf-project.org/
[3] vCard Ontology for describing People and organizations. https://www.w3.org/TR/vcard-rdf/
[4] A web platform for text-mining. https://gargantext.org of the Institute of Complex Systems (Paris).

Table 1. The first fifteen compound multi-words and their occurrences extracted by data mining with the keyword "OAEI" in a corpus of two hundred documents.

Multi-word extracted	Occurrence
ontology matching	59
semantic web	42
ontology mapping	39
ontology alignment	38
different ontology	36
data sources	16
cheminform abstract	16
ontology alignment evaluation initiative	16
open data	15
schema matching	14
schema matcher	14
semantic interoperability	12
matching process	11
large ontologies	11
similarity measures	10

into the text mining process to enter a virtuous circle. We stoped the process when we considered the list of terms extracted by the text mining stops evolving.

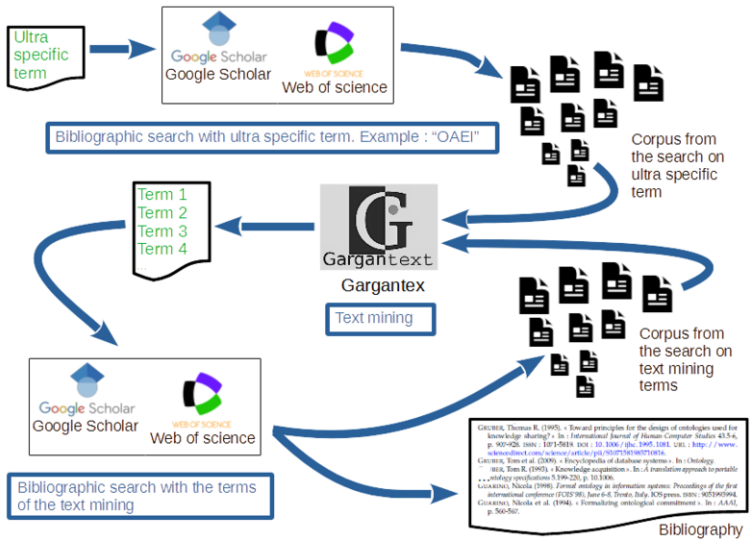

Figure 3. Methodology for bibliographic search and enrichment.

We have thus obtained a first set of relevant documents in the field of data linking and its different techniques. It is from this first corpus that we will subsequently extract a list of five articles dealing with the problems of heterogeneity.

4.2. Review of articles addressing different heterogeneity issues

We classify the different forms of heterogeneity found in the literature according to our four issues in order to continue our classification of LPTs. To do so, we manually re-

viewed the articles dealing with heterogeneities compiled in the corpus explained above. These articles are either about Instance Matching (IM) where one consider assertions (notion of instance) or about Ontology Matching (OM) where one deal with the reconciliation of models (notion of classes). Although distinct, these two domains are complementary in the execution of data linking tasks. We have finally selected the following five articles for their relevance and their global vision on the subject of heterogeneity:

- Klein [21] proposes a classification of different heterogeneities encountered in the combined use of independently constructed ontologies. This work identifies three main families of heterogeneities:
  - Heterogeneity related to practice (e.g., non application of language tags, input errors, duplicates).
  - Heterogeneity linked to the mismatch of languages used to express these ontologies. At this level, a distinction is made between heterogeneities related to the linguistic level (e.g.,syntax, representation, semantics and expressivity) and those related to the ontological level (e.g., paradigm, concept description, coverage of model, synonymy).
  - Heterogeneity related to the versioning of one or more of the ontologies involved.
- Bergman [22] addresses the issue of resolving semantic heterogeneities in the context of using the semi-structured XML language (based largely on the work of Pluempitiwiriyawej and Hammer [23]) and, by extension, RDF and ontology representation languages like OWL. He considers that even within an identical domain there will always be different "world views" as long as independent teams create ontologies due to the flexibility of semi-structured schemas. Moreover, during serialization, XML files and ontologies can be confronted with syntax or structure problems. This work identifies four categories of causes for these heterogeneities:
  - Heterogeneity related to structure. This occurs when the schemas of the sources that represent related or overlapping data do not match (e.g., first and last name aggregation).
  - Heterogeneity related to domain. This occurs when the semantics of the data sources are different (e.g., Different scales and units of measurement).
  - Heterogeneity related to data. This occurs when there are discrepancies between the values of similar or related data (e.g., spelling mistakes).
  - Heterogeneity related to language.This occurs when there are differences in the encoding and use of different languages (e.g., Use of French and English).

  Bergman estimates there are more than forty discrete categories of heterogeneity. As our work is focused on RDF datasets, some of the heterogeneities described in the context of the use of XML seemed irrelevant (e.g., the notion of element order which is non-existent in RDF). Of the forty or so heterogeneities presented, we have selected twenty-six which fall into the four main categories.
- Euzenat and Shvaiko's work [24] is related to OM rather than IM; but still bring in an interesting analysis of heterogeneities that we can apply to data linking. They consider the following four main types of heterogeneities:
  - Syntactic heterogeneity: ontologies are expressed in different representation languages.
  - Terminological heterogeneity: ontologies have variations in naming objects (car vs. automobile).
  - Conceptual heterogeneity: ontologies have differences in modeling choices for the same domain. They can be differences in coverage, granularity or perspective.

  - Semiotic heterogeneity: ontologies describe the same thing (e.g., a sharp metal blade with a handle) that people/users will interpret differently depending on the context (a knife can be a weapon or a kitchen utensil).
- Achichi et al. [25] pragmatically classify the heterogeneities encountered by the designers of data linking tools as:
  - Value dimension: for heterogeneity problems at the level of terminology, language used and distinction between datatype properties and object properties.
  - Ontological dimension: for the problems of heterogeneity of vocabularies, structures, property depth, description and key.
  - Logical dimension: for class and property heterogeneity problems.
  - Data quality dimension: for the problems of transgression of good practice, heterogeneity of value type or non-updated dataset.
- Assi et al. [26] address the issue of IM. They introduce the scalability problem when it comes to IM on large datasets. Plus, they classify the heterogeneities as:
  - Value heterogeneity: gathering the notions of multilingualism, data format and data quality.
  - Structural heterogeneity: gathering the notions of vocabulary heterogeneity, predicate level and predicate granularity.
  - Logical heterogeneity: gathering the notions of hierarchical variation.

We found many similarities between these different heterogeneities, both in terms of organization by level and detail, but also many differences. We justify this diversity by the fact that the domains covered are not necessarily identical. For example, instance matching and ontology matching and because the levels of detail of each studies is different. Through these five articles, we were able to identify 69 descriptions of heterogeneity (See Table 2).

Table 2. Number of heterogeneity descriptions by authors.

Author(s)	Number of heterogeneity descriptions	Reference
Klein	11	[21]
Bergman	26	[22]
Euzenat and Shvaiko	6	[24]
Achichi et al.	13	[25]
Assi et al.	13	[26]

In order to better refer to them, we established a summary fact-sheet for each type of heterogeneity encountered, identified by a token. The tokens have been colored according to the authors who report them, as shown in Figure 4 presenting one of these summary fact-sheet.

4.3. An iterative methodology

In Section 3, we introduced four issues to evaluate how well they correspond to the various types of heterogeneities discussed in the paper corpus. To incorporate these heterogeneities into our new classification, we conducted manual clustering iterations. An example of this process is illustrated in Figure 5. In each iteration, we categorized the tokens into different themes based on their authors' descriptions to refine our classification. Some tokens were found in multiple clusters in subsequent iterations –as in the case where Assi et al. [26] mentioned that "the incorrectness simply refers to the

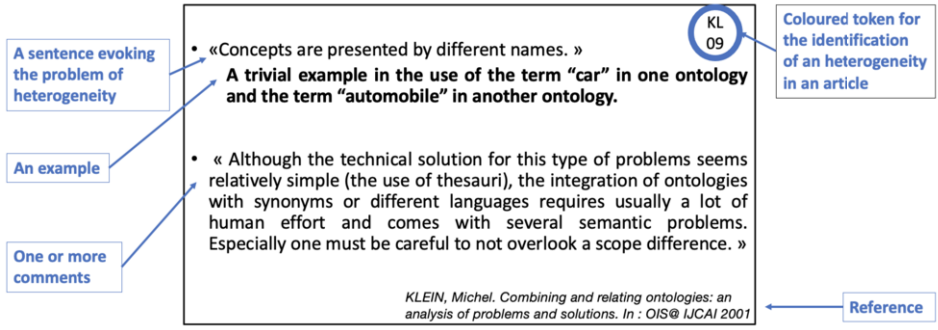

Figure 4. fact-sheet on synonymy problems of concept names according to Klein.

data typographical errors"– which can affect the value of a predicate as well as on the predicate itself. As a result of the first iteration, we were unable to classify some heterogeneities, which prompted us to establish a fifth primary level called "Problem at graph level". For example, when Achichi et al. [25] talks about key heterogeneity: "a property used to provide individual identifiers specific to a dataset, for example the identifiers of bibliographic entries in two libraries. In both cases, the values of these key properties are not comparable from one dataset to another." We could not classify this problem of heterogeneity within any of our four initial issues.

After four iterations, we arrived at a final classification that addresses all heterogeneity issues, organized into five primary levels based on the heterogeneity descriptions found in the literature.

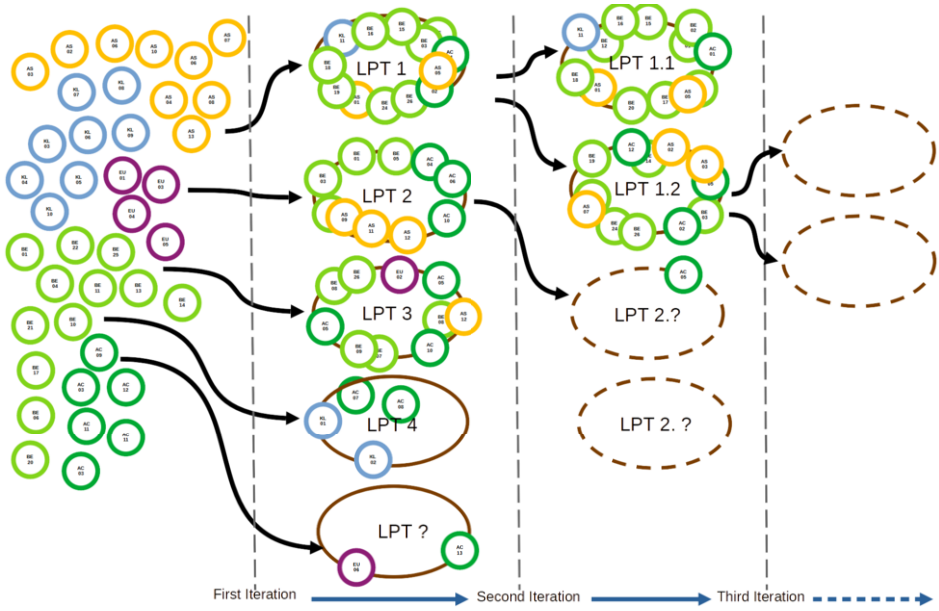

Figure 5. Schema of our iterative approach to develop the classification of LPTs, based on the reported heterogeneities in the selected articles. Each token in the figure represents a heterogeneity described in an article and is associated with a specific author and colour-coded accordingly.

5. Results

5.1. A classification for Linking Problem Types (LPTs)

We present here the results of the classification process previously explained. At the first hierarchy level, we find the four intuitive groups of problems to which we have added another group (i.e.,"Problem at graph level") to capture problems related to the nature of the graphs (see Figure 6). In this Figure and the following ones, the colored pie charts represent the different distributions of heterogeneities described by each of the previously selected authors.

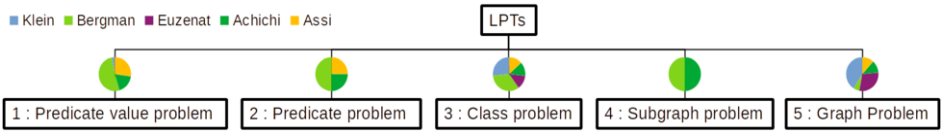

Figure 6. First level of the hierarchical classification of Linking Problem Types (LPTs).

5.1.1. Predicate value problems

Predicate value problems can be divided into terminological problems on one side and structural problems on the other (Figure 7). At the terminological level, the classification extends over three levels of granularity expressing at the finest level the problems of synonymy, homonymy or language reported in the literature mainly by Bergman [22] and Assi et al. [26].

A problem that would fall within the scope of LPT 1.1.2.5 would be, for example, a pair of datasets in which the data would be inconsistent (e.g., New York City would be described with 8,804,190 inhabitants on one side and 8,800,000 inhabitants on the other). For LPT 1.1.3, an example would be, a dataset pair in which literals do not have language labels (e.g., ,"barbecue" instead of "barbecue@en"), which would prevent automatic determination of the label language. For LPT 1.2.2, an example would be, a dataset pair in which the city of New York is represented as the object of a triple by the literal "New York"@en on the one hand, and by its URI, `https://www.wikidata.org/wiki/Q60` pointing to the corresponding Wikidata page on the other.

5.1.2. Predicate problems

Predicate problems can be divided into predicate terminological problems, predicate structural problems (as predicate value problems) and predicate vocabulary problems (see Fig. 8).

An issue within the scope of LPT 2.1.5 would be, for example, a pair of datasets where the same predicate has a typing error (e.g., `hasPopulation` on one side and `hasPupoltion` on the other). For LPT 2.2.3, an example would be, a dataset pair in which a extra node (which can be a blank node) must be inserted or deleted in order to retrieve the same information (e.g., `New York hasNikeName Big Apple` on one side and `New York isCalled _b1 hasNikeName Big Apple ; New York isCalled _b1 hasAcronym NYC` on the other). For LPT 2.3 an example would be, a dataset pair which would express the same information using predicates from different vocabularies (e.g., `foaf:name` on one side and `rdfs:label` on the other).

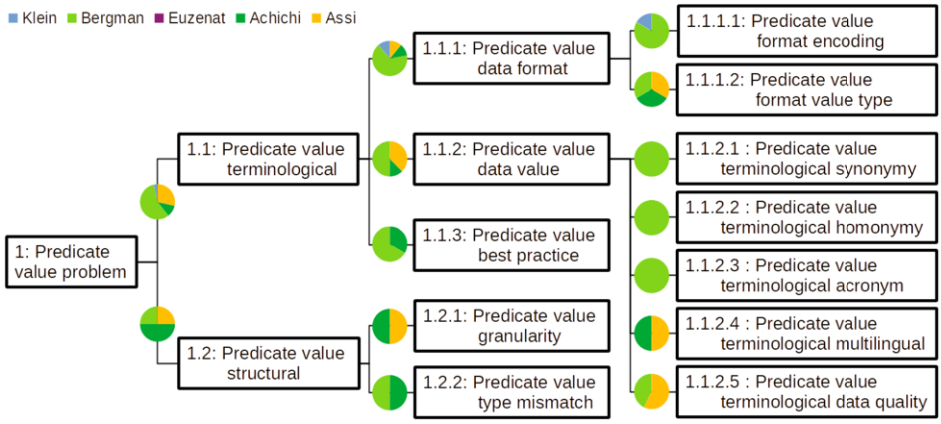

Figure 7. Part of the classification of LPTs: Predicate value problems.

At the level of granularity, we notice that the final levels of the classification are only described by a few authors. For example, only Bergman describes in detail the heterogeneities associated with terminological synonymy, homonymy and acronymy.

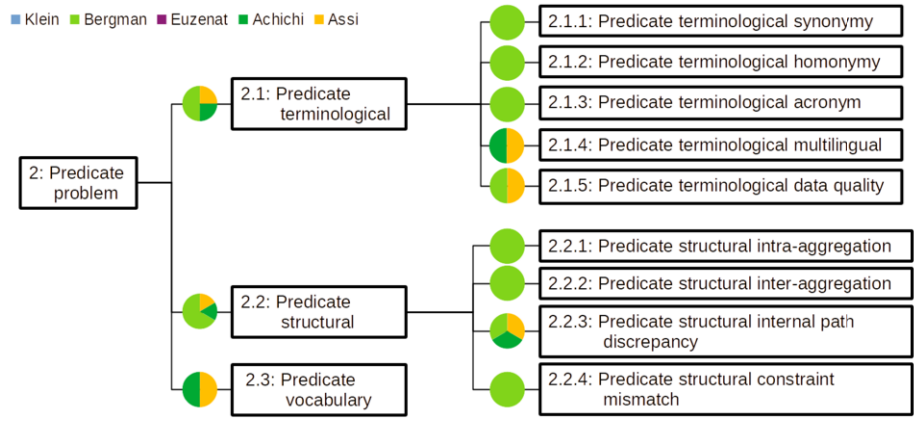

Figure 8. Part of the classification of LPTs: Predicate problems.

5.1.3. Class problems

Class problems can be divided into class terminological problems on one side and specialization/generalization on the other (see Fig. 9). We find a clustering around terminological problemes, which seems normal, if one consider it is a special case of a predicate value problem. Another grouping appears around the specialization/generalization problem more specific to the class domain reported by Klein [21], Bergman [22] and Achichi et al. [25].

For LPT 3.1.4 an example would be, a dataset pair in which there are variations in names for the same concept (e.g., `Paper` on one side and `Article` on the other). A example for LPT 3.2 would be, a dataset pair in which more general or specific concept are ised (e.g., `Phone` on one side and `HomePhone` or `Smartphone` on the other).

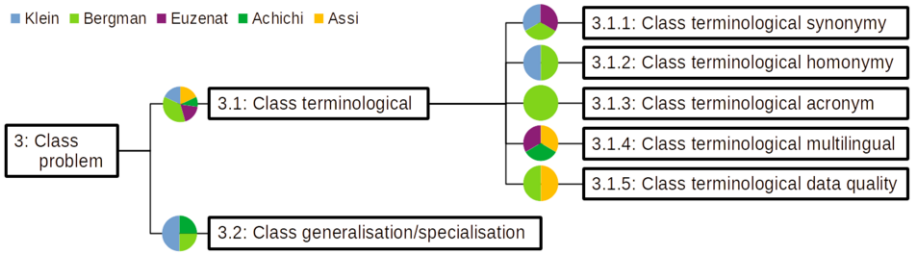

Figure 9. Part of the classification of LPTs: Class problems.

5.1.4. Subgraph problems

Subgraph problems can be divided into subgraph descriptive heterogeneity problems on one side and subgraph no textual description problems on the other (see Fig. 10). We make a distinction between the heterogeneity of description and the absence of description of certain characteristics of the entity.

For LPT 4.1, an example would be a pair of datasets describing a resource with a different amount of information (e.g., the city of New York with its name, population and geographic location on one side and its name and location on the other).

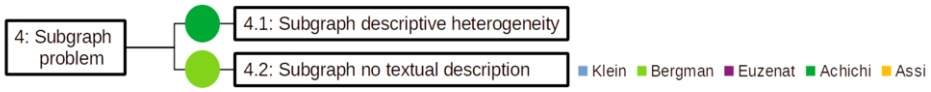

Figure 10. Part of the classification of LPTs: Subgraph problems.

5.1.5. Graph problems

Graph problems can be divided into eight levels (see Fig. 11). Graph problems level had to be explicitly added to the classification, because, we distinguish more general problems from those presented at the RDF triplet level. Problems like scalability and expressiveness of some languages compared to others concerning for example the expression of negation.

For LPT 5.4.4, an example would be a pair of datasets that use distinct languages with differences in the representation of logical notions (e.g., a language that directly expresses class disjunctions (A disjoins B) on the one hand and a language requiring the use of negation (A subclass-of (NOT B) on the other). For LPT 5.4.6, an example would be a pair of datasets describing the population and dynamics of the same city but at different times. For LPT 5.6, an example would be a pair of datasets where at the level of the graphs the description patterns would be identical whereas they would be different descriptions (e.g., two sets of triples composed only of individuals of the class person but with on one side a single reflexive relation ”hasSister” and on the other a single reflexive relation ”hasBrother”). This type of problem will be especially useful for the embedding graph.

Once again, we note some differences in the distribution of each problems by level as done by the authors, such as Achichi, who only reports heterogeneities related to the heterogeneity of graph conceptual keys and the timeliness of graph conceptual datasets.

We added the LPT 5.8, as we think it could be useful in the development of hybrid techniques mixing IM and OM where the absence of TBox would be perceived as a problem.

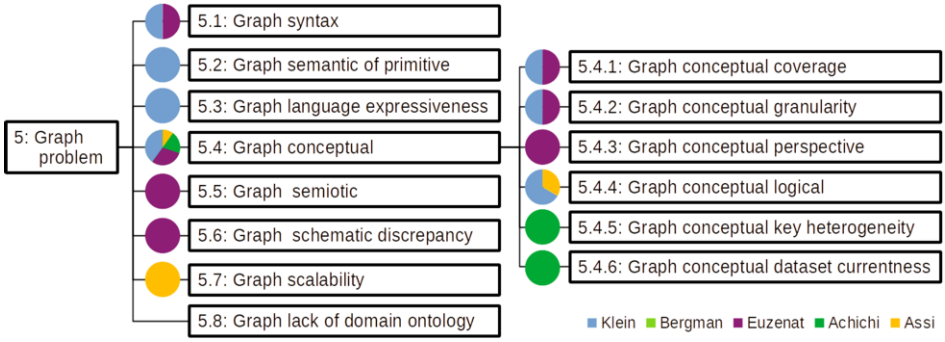

Figure 11. Part of the classification of LPTs: Graph problems.

5.2. *Formalization of the LPT classification*

We formalized the LPT classification, as illustrated in Figure 12. We use the `lpt` prefix as the namespace for our classification. The `lpt:LPT` class is the primary class in our model. To maximize reuse, we rely on established vocabularies as much as possible instead of creating new classes and properties. Especially, we used SKOS, RDF-S, Dublin Core and PROV-O [27] to describe the classes. Additionally, we used the DCAT vocabulary [28] to define datasets and their distributions. We introduced the class `lpt:PairOfDatasets` to represent a couple of `dcat:Datasets` that is or need to be linked. This class reifers the pair into an object that can be described on its own e.g., status of linking, date of linking, source of linking; we do not describe these here. The `lpt:occursIn` property is the key relation in our model: it encodes the fact that a certain linking problem type occurs/appears in a certain pair of datasets. To provide a detailed description of such a pair, instances of the `lpt:PairOfDatasets` class are linked to two separate individuals of the `dcat:Dataset` class using the `lpt:hasSource` and `lpt:hasTarget` properties. The property `prov:wasInfluencedBy` connects the `lpt:LPT` class to the underlying heterogeneities reported by various authors and encoded with the class `lpt:Heterogeneity`. The bibliographic sources from which we derived the descriptions of the heterogeneities that guided our classification are captured with the property `prov:wasDerivedFrom` to an object in the BIBO ontology [29].

An example of instantiation of the LPT classification model is provided in Figure 13. This example is in fact a real example of the appearance of the LPT 2.1.1 problem called `Predicate terminological synonymy` in the `lpt:datasetOAEI101` and `lpt:datasetOAEI205` datasets accessible from https://oaei.ontologymatching.org/tests/101/onto.rdf and https://oaei.ontologymatching.org/tests/205/onto.rdf respectively. This pair of RDF datasets is made available by OAEI to allow future participants to test their tools.

This classification is currently being made available on the web in OWL format. The choice of this representation language was made with a view to encoding a hierarchical classification for future use by software solutions exploiting the inference capabilities

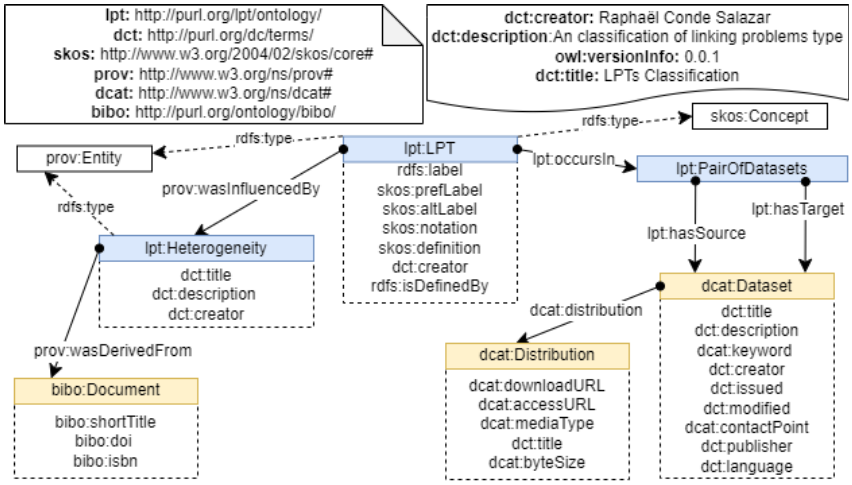

Figure 12. Conceptual model for defining Linking Problem Types.

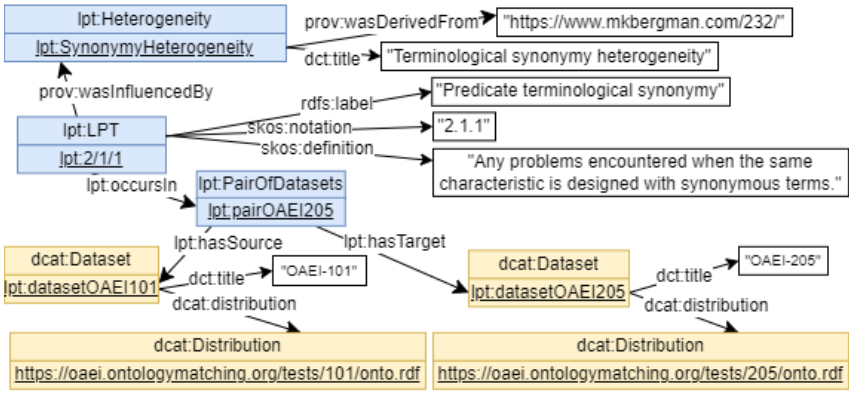

Figure 13. An example of formalization of a LPTs in RDF.

provided by this language. This hierarchisation based on the `rdfs:subClassOf` property involving classes for each LPT (e.g., `LPT_1_1_2 rdfs:subClassOf LPT_1_1`) is not represented here for lack of space.

6. Conclusion

Data linking allows similar entities to be linked, so that semantic data spread over several heterogeneous datasets can be used more effectively. In this paper, we therefore propose a formalized classification of the different types of problems that can be encountered when linking RDF datasets. We hope that this classification will help the data linking community to better identify the problems that may arise when two RDF datasets with heterogeneous terminology, structure, and logic need to be linked. Establishing a precise profile, as close as possible to the RDF data to be processed, should allow a better choice of the algorithmic module(s) needed to solve a data linking task, in an attempt

to improve the performance of existing data linking tools, most of which use generic solutions.

In the future, we plan to continue to develop our classification by continuing to provide, for example, for each LPT described, examples of real cases from pairs of datasets used during data linking competitions. The different techniques capable of solving these LPTs will also be attached. Ultimately we want to make an OWL ontology that we will of course make available online for all users in the field of data linking.

This work is achieved in the context of the DACE-DL (Data-Centric AI-driven Data Linking) project [5] which proposes a paradigm shift in data linking by focusing on a bottom-up, data-centric methodology [11]. The objective of this research project is to use machine learning techniques and representation learning models to improve data linking by facilitating the application of the right linking tool to the relevant linking problem. Thus the need to formalize a classification of linking problem types. We therefore envision our classification to be used to determine, via learning processes, the relevant specific linking tool modules necessary for data linking according to the different problems exposed by a pair of datasets in order to provide a more specific solution to a linking task than current approaches.

Another perspective of this work, is to experiment an unsupervised machine learning process to categorize different pairs of datasets for which we would have manually determined the different LPTs potentially exposed. The goal of such experimentation would be to verify our grouping operated via the LPT classification can be corroborated by a categorization performed via a machine learning process. These dataset pairs are taken from various datalinking benchmarks such as OAEI. Each pair is documented with the different LPTs they expose, an additional file containing the different alignments that should theoretically be obtained after running a linking tool and the linking tool that has been tested as the best performing.. Other information on these datasets is provided (i.e. description, year of creation, origin, type of alignment(T-Box/Schema matching, Instance matching or link discovery, Instance and schema matching and Tabular data to Knowledge Graph matching). In our next project, we aim to set up an automated software solution that would receive as input a pair of datasets that we are trying to link and as output the LPTs that they expose.

Acknowledgement

This work has been supported by the DAta-CEntric AI-driven Data Linking project (DACE-DL - https://dace-dl.github.io/ - ANR-21-CE23-0019).

References

[1] Berners-Lee T, Hendler J, Lassila O. The semantic web. Scientific american. 2001;284(5):34-43.
[2] Beek W, Raad J, Wielemaker J, van Harmelen F, editors. sameas. cc: The closure of 500m owl: sameas statements. Springer; 2018.
[3] Correndo G, Penta A, Gibbins N, Shadbolt N, editors. Statistical analysis of the owl: SameAs network for aligning concepts in the linking open data cloud. Springer; 2012.
[4] Halpin H, Hayes PJ, McCusker JP, McGuinness DL, Thompson HS, editors. When owl: Sameas isn't the same: An analysis of identity in linked data. Springer; 2010.
[5] Ding L, Shinavier J, Shangguan Z, McGuinness DL, editors. SameAs networks and beyond: Analyzing deployment status and implications of owl: sameAs in linked data. Springer; 2010.

[5] Data-Centric AI-driven Data Linking. https://dace-dl.github.io/

[6] Nentwig M, Hartung M, Ngonga Ngomo AC, Rahm E. A survey of current link discovery frameworks. Semantic Web. 2017;8(3):419-36.
[7] Euzenat J, Meilicke C, Stuckenschmidt H, Shvaiko P, Trojahn C. Ontology alignment evaluation initiative: Six years of experience. In: Journal on data semantics XV. Springer; 2011. p. 158-92.
[8] Algergawy A, Faria D, Ferrara A, Fundulaki I, Harrow I, Hertling S, et al., editors. Results of the ontology alignment evaluation initiative 2019. vol. 2536; 2019.
[9] Abd Nikooie Pour M, Algergawy A, Amini R, Faria D, Fundulaki I, Harrow I, et al., editors. Results of the ontology alignment evaluation initiative 2020. vol. 2788. RWTH; 2020.
[10] Pour M, Algergawy A, Amardeilh F, Amini R, Fallatah O, Faria D, et al., editors. Results of the ontology alignment evaluation initiative 2021. vol. 3063. CEUR; 2021.
[11] Todorov K, editor. Datasets First! A Bottom-up Data Linking Paradigm; 2019.
[12] Euzenat J. Extraction de clés de liage de données (résumé étendu). In: 16e conférence internationale francophone sur extraction et gestion des connaissances (EGC). Hermann; 2016. p. 9-12.
[13] Atencia M, David J, Scharffe F, editors. Keys and Pseudo-Keys Detection for Web Datasets Cleansing and Interlinking. Springer; 2012.
[14] Symeonidou D, Armant V, Pernelle N, Saïs F, editors. Sakey: Scalable almost key discovery in RDF data. Springer; 2014.
[15] Symeonidou D, Armant V, Pernelle N. BECKEY: Understanding, comparing and discovering keys of different semantics in knowledge bases. Knowledge-Based Systems. 2020;195:105708.
[16] Babic B, Nesic N, Miljkovic Z. A review of automated feature recognition with rule-based pattern recognition. Computers in industry. 2008;59(4):321-37.
[17] Käfer T, Harth A. Rule-based Programming of User Agents for Linked Data. LDOW@ WWW. 2018;2073.
[18] Wang Q, Mao Z, Wang B, Guo L. Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering. 2017;29(12):2724-43.
[19] Dai Y, Wang S, Xiong NN, Guo W. A survey on knowledge graph embedding: Approaches, applications and benchmarks. Electronics. 2020;9(5):750.
[20] Harispe S, Ranwez S, Janaqi S, Montmain J. Semantic similarity from natural language and ontology analysis. Synthesis Lectures on Human Language Technologies. 2015;8(1):1-254.
[21] Klein M, editor. Combining and relating ontologies: An analysis of problems and solutions; 2001.
[22] Bergman M. Sources and classification of semantic heterogeneities. Web Blog: AI3-Adaptive Information, Adaptive Innovation, Adaptive Infrastructure. 2006.
[23] Pluempitiwiriyawej C, Hammer J. A classification scheme for semantic and schematic heterogeneities in XML data sources. TR00-004, University of Florida, Gainesville, FL. 2000.
[24] Euzenat J, Shvaiko P. Ontology matching. vol. 18. Springer; 2007.
[25] Achichi M, Bellahsene Z, Ellefi MB, Todorov K. Linking and disambiguating entities across heterogeneous RDF graphs. Journal of Web Semantics. 2019;55:108-21.
[26] Assi A, Mcheick H, Dhifli W. Data linking over RDF knowledge graphs A survey. Concurrency and Computation: Practice and Experience. 2020;32(19):e5746.
[27] Lebo T, Sahoo S, McGuinness D, Belhajjame K, Cheney J, Corsar D, et al. Prov-o: The prov ontology. W3C recommendation. 2013;30.
[28] Albertoni R, Browning D, Cox S, Beltran AG, Perego A, Winstanley P. Data catalog vocabulary (DCAT)-version 2. World Wide Web Consortium. 2020.
[29] D'Arcus B, Giasson F. Bibliographic ontology specification. Madrid: Biblioteca Nacional Española. 2009.