A Business Intelligence Tool for Explaining Similarity*

Simona Colucci[1], Francesco M. Donini[2], Nicola Iurilli[1], and Eugenio Di Sciascio[1]

[1] Politecnico di Bari, Bari, Italy
simona.colucci@poliba.it, n.iurilli@studenti.poliba.it, eugenio.disciascio@poliba.it
[2] Università degli Studi della Tuscia, Viterbo, Italy
donini@unitus.it

Abstract. Agile Business often requires to identify similar objects (firms, providers, end users, products) between an older business domain and a newer one. Data-driven tools for aggregating similar resources are nowadays often used in Business Intelligence applications, and a large majority of them involve Machine Learning techniques based on similarity metrics. However effective, the mathematics such tools are based on does not lend itself to human-readable explanations of their results, leaving a manager using them in a "take it as is"-or-not dilemma. To increase trust in such tools, we propose and implement a general method to explain the similarity of a given group of RDF resources. Our tool is based on the theory of Least Common Subsumers (LCS), and can be applied to every domain requiring the comparison of RDF resources, including business organizations. Given a set of RDF resources found to be similar by Data-driven tools, we first compute the LCS of the resources, which is a generic RDF resource describing the features shared by the group recursively—*i.e.*, at any depth in feature paths. Subsequently, we translate the LCS in English common language. Being agnostic to the aggregation criteria, our implementation can be pipelined with every other aggregation tool. To prove this, we cascade an implementation of our method to *(i)*the comparison of contracting processes in Public Procurement (using TheyBuyForYou), and *(ii)* the comparison and clustering of drugs (using k-Means) in Drugbank. For both applications, we present a fairly readable description of the commonalities of the cluster given as input.

Keywords: Explainable Artificial Intelligence (XAI), Resource Description Framework (RDF), Least Common Subsumer (LCS)

* This preprint has not undergone peer review (when applicable) or any post-submission improvements or corrections. The Version of Record of this contribution is published in "MOBA 2022: Model-Driven Organizational and Business Agility", and is available online at https://doi.org/10.1007/978-3-031-17728-6_5".

1 Introduction

The ability of a business organization to rapidly adapt to changing conditions—referred to as Agility—often requires, among other needs, a rapid identification of pairs—or clusters—of similar objects, being they partners, products, requirements, etc. This problem has been widely investigated in the literature and applied to heterogeneous application domains, ranging from business strategy [20], to manufacturing [10] to drug analysis [9], among others.

Aggregating similar entities in a large dataset may reveal patterns, point out special groups, and in general it helps in understanding data. Data-driven tools for aggregating resources by some similarity measure are nowadays available by the dozens [7, 14], and a large majority of them involve Machine Learning techniques based on similarity metrics. However, the mathematics such tools are based on is rather complex, therefore approaches that make the tool transparent in order to explain a user how the cluster was constructed do not lead to human-readable explanations. This problem leaves to the manager using the tool the burden to make explicit which characteristics are similar in the aggregated resources—a task which is not always self-evident.

We provide a method for describing the commonalities among a given set of RDF resources, that can be pipelined to any tool for clusterization whose output is one or more cluster of RDF resources judged similar by the tool. Since our method is logic-based, it can abstract with blank nodes features that, although different, lead to a common value through a recursive chain (see Section 3 for more details).

As a byproduct, our method can be used also in a fine-tuning phase of a clusterizing tool, since it allows one to immediately identify which clusters are significant—among the ones obtained—for a human reader and possibly which features in the data lead to significant clusterization.

The rest of this paper is organized as follows: in the next section, we report related work on Natural Language Generation for RDF. Section 3 recalls main notions about the definition of LCS in RDF. In Section 4, we show our logic-based method for the explanation of the similarity of RDF resources. We demonstrate the feasibility of our approach by showing its results in two different contexts in Section 5. Conclusions close the paper.

2 Related Work

The need for making explicit the interpretation of clustering results emerged a long time ago. In 1980, in fact, conceptual clustering [12] was introduced as the problem of returning clusters of resources, together with a concept explaining the proposed aggregation. In the meanwhile, several conceptual clustering approaches and algorithms have been proposed, the most influential of which have been reviewed in a recent work by Pérez-Suárez *et al.* [13]. None of the approaches summarized by Pérez-Suárez *et al.* deals with RDF resources. An approach to conceptual clustering of RDF resources based on LCS was proposed by Colucci *et al.* [6].

Our proposal stems from a similar need, but, differently from works in the field of conceptual clustering, we do not build clusters of RDF resources; we focus only on their explanation. To this aim, we propose a logic-based methodology: the natural language text explaining the cluster is generated from the set of RDF triples computed as the LCS of all cluster items, which abstracts the commonalities of all resources.

The problem of generating Natural Language text from Semantic Web (SW) data has been widely addressed in the literature and continues to represent a relevant research topic. A systematic review of main approaches up to 2014 was conducted by Bouayad-Agha *et al.* [1] who classify at least 11 Natural Language Generation (NLG) approaches working on RDF graphs, w.r.t. to several features, including the verbalization request (part of the input graph to verbalize) and the communicative goal (information to return). Possible communicative goals are: returning all facts in the verbalization request, returning a user-selected set of facts, returning the most typical facts, returning the most relevant facts. Apparently, no research work is able to generate text from derived triples not explicitly stated in RDF and to manage anonymous resources. Bouayad-Agha *et al.* [1] also point out the need for summarizing information among challenging communicative goals.

The current research trend, to the best of our knowledge, is mostly focused on improving the readability of textual descriptions generated from RDF, w.r.t. criteria set as baselines.

In particular, the WebNLG challenge [3] significantly boosted the proposal of research solutions in NLG, by providing a benchmark corpus of English sentences verbalizing RDF triples. The challenge has been repeated in 2020 [21], including a larger corpus in both English and Russian and adding the subtask of parsing natural language text in RDF. The current dataset refers to 15 categories of resources. The object of the challenge is, again, improving the performance in the generation of baseline sentences rather than proposing forms of verbalization lending to richer explanation.

Traditionally, NLG approaches have been based on rules and templates (see [2], among others), that make such solutions highly domain-dependent and demanding manual intervention.

Recently, the advancements in deep learning have opened the way to neural network-based NLG models. Among them, the Sequence to Sequence (SEQ2SEQ) framework [17] has been employed by Vougiouklis *et al.* [18] to propose a framework, Neural Wikipedian, to generate summaries of RDF triples. The approach is able to summarize triples involving the same entity either as a subject or as an object, but, again, only explicitly stated facts are verbalized and anonymous resources are not managed.

Also the Neural Entity Summarization of Li *et al.* [11] collects only triples that are already present in the RDF descriptions, without handling blank nodes.

Differently from the above approaches, we propose a template-based method that can use blank nodes to abstract several triples with common predicate/object, and, more importantly, can chain triples with blank nodes that eventually reach

the same known object (see the example about contracting processes in Sect.5.1). The informative potential of our method is in the logic-based computation of the RDF graph to verbalize: a rooted graph summarizing in triples the commonalities shared by groups of RDF resources. We show in the rest of the paper how this method for NLG may support the explanation of RDF similarity.

3 LCS in RDF

To make this paper self-contained, we briefly recall here the definition of Least Common Subsumer (LCS) from works by Colucci *et al.* [5, 4], along with some preliminary notions. First of all, to compare specific resources r, s in RDF, we need the definition of *rooted* RDF-*graph* (in brief *r-graph*): a pair $\langle r, T_r \rangle$ which isolates resource r inside the RDF-graph T_r. Secondly, $G[s \rightarrow t]$ denotes the graph obtained from G by substituting each occurrence of s with t. Then, the definition of Simple Entailment $T_r \models T_s$ [8] between two RDF-graphs T_r, T_s, is extended to r-graphs as follows [4, Def.6]:

Definition 1. *[Rooted Entailment] Let $\langle r, T_r \rangle, \langle s, T_s \rangle$ be two r-graphs. We say that $\langle r, T_r \rangle$* entails *$\langle s, T_s \rangle$—denoted by $\langle r, T_r \rangle \models \langle s, T_s \rangle$—in exactly these cases:*

1. *if s is a blank node, then*
   (a) *if r is not a blank node, $T_r \models T_s[s \mapsto r]$ must hold;*
   (b) *if also r is a blank node, then $T_r[r \mapsto u] \models T_s[s \mapsto u]$ must hold for a new URI u occurring neither in T_r nor in T_s;*
2. *otherwise (i.e., s is not a blank node), if $s = r$, then $T_r \models T_s$ must hold.*

In all other cases (i.e., s is not a blank node and $s \neq r$), $\langle r, T_r \rangle$ never entails *$\langle s, T_s \rangle$.*

Intuitively, Rooted Entailment extends Simple Entailment with the requirement that the root of a graph is mapped to the root of the other. When both resources r, s are URI, this is possible only when $s = r$ (Case 2), while when either r or s is a blank node (Cases 1b and 1a), the mapping is enforced by a suitable substitution.

Rooted Entailment is at the basis of the definition a Common Subsumer (CS) of two r-graphs $\langle a, T_a \rangle$, $\langle b, T_b \rangle$:

Definition 2 (Common Subsumer, [4, Def.7]). *Let $\langle a, T_a \rangle$, $\langle b, T_b \rangle$ be two r-graphs. An r-graph $\langle x, T_x \rangle$ is a Common Subsumer (CS) of $\langle a, T_a \rangle$, $\langle b, T_b \rangle$ iff both $\langle a, T_a \rangle \models \langle x, T_x \rangle$ and $\langle b, T_b \rangle \models \langle x, T_x \rangle$.*

Finally, a Least Common Subsumer (LCS) of two RDF resources can be defined as follows:

Definition 3 (Least Common Subsumer [4, Def.8]). *Let $\langle a, T_a \rangle$, $\langle b, T_b \rangle$ be two r-graphs. An r-graph $\langle x, T_x \rangle$ is a Least Common Subsumer (LCS) of $\langle a, T_a \rangle$, $\langle b, T_b \rangle$ iff both conditions below hold:*

1. *$\langle x, T_x \rangle$ is a CS of $\langle a, T_a \rangle$, $\langle b, T_b \rangle$;*

2. *for every other CS* $\langle y, T_y\rangle$ *of* $\langle a, T_a\rangle$, $\langle b, T_b\rangle$*:*
   if $\langle y, T_y\rangle \models_{\mathcal{R}} \langle x, T_x\rangle$ *then* $\langle x, T_x\rangle \models_{\mathcal{R}} \langle y, T_y\rangle$*, (i.e.,* $\langle x, T_x\rangle$ *and* $\langle y, T_y\rangle$ *are equivalent under Simple Entailment).*

Colucci *et al.* [4] proved that an LCS of two r-graphs is unique—up to blank renaming—so we can talk about "the" LCS. Moreover, the LCS enjoys the following properties:

- Idempotency: $LCS(\langle a, T_a\rangle, \langle a, T_a\rangle) = \langle a, T_a\rangle$
- Commutativity: $LCS(\langle a, T_a\rangle, \langle b, T_b\rangle) = LCS(\langle b, T_b\rangle, \langle a, T_a\rangle)$
- Associativity:
  $LCS(\langle a, T_a\rangle, LCS(\langle b, T_b\rangle, \langle c, T_c\rangle)) = LCS(LCS(\langle a, T_a\rangle, \langle b, T_b\rangle), \langle c, T_c\rangle)$.

Associativity relies on a fundamental property of LCSs: the LCS of two r-graphs is itself an r-graph, so it can be used as the argument of another LCS operation with a third r-graph, and so forth. Associativity ensures that the order in which resources are taken—when computing the LCS of all of them—does not matter.

We also recall some definitions modifying the basic notions of Graph Theory to RDF-graphs [4], used in the rest of the paper. First, an RDF-*path* from r to s is a sequence of triples $t_1, \ldots, t_n$ in which the subject of t_1 is r, *either* the predicate *or* the object of t_n is s, and for $i = 1, ..., n-1$, either the predicate or the object of t_i is the subject of t_{i+1}. A resource r is RDF-*connected* to a resource s if there exists an RDF-path from r to s. The *length* of such an RDF-path is n, and the RDF-*distance* between two resources is the length of the shortest RDF-path between them. Also, the RDF-distance between a resource r and a triple t is the shortest RDF-distance between r and the subject of t—in particular, triples which r is the subject of, have zero-RDF-distance from r itself, as expected.

We propose to use the LCS of a cluster of RDF resources—where the cluster can be obtained in any way—to explain their commonalities. To this end, we attached to the construction of an LCS its verbalization in English common language, as described in the next section.

4 From LCS r-graphs to NLG Explanation

Our approach generates a verbal explanation of the similarity of groups of RDF resources, starting from the triples in their LCS.

Real applications managing RDF resources need some preliminary choices to ensure feasibility. In fact, RDF-based applications cannot take all triples describing a resource r into account, given the huge and always increasing dimensions of available datasets. Thus, it is crucial to select which triples qualify r and build its r-graph. Colucci *et al.* [4] proposed explicit criteria for this choice:

1. data sources: which datasets (one or more) to explore for the comparison;
2. RDF-distance: exclude triples which are "too far" from r;
3. stop-patterns: exclude triples which fit a given pattern $\ll s\ p\ o \gg$;

4. connectedness: there must be an RDF-path from r to the subject of each chosen triple.

Notably, the first three choice criteria can be parameterized for the particular application at hand.

Still, the computed LCS may contain too many triples which, although logically implied by the r-graphs of all analyzed resources, provide little information. Colucci *et al.* name these triples *uninformative triples*, and propose to eliminate them from the comparison result. The result is a—no more Least—Common Subsumer, containing only the most informative triples deducible from all r-graphs.

The set of stop-patterns and uninformative triples used in this paper include both general patterns/triples (to be discarded in every application domain), and some domain-dependent patterns/triples, defined through the analysis of our results.

We propose a template-based NLG tool for explaining the content of a CS we consider significant for the similarity analysis at hand. In fact, the tool allows developers to flexibly set the context of analysis through the specification of the list of uninformative triples, the datasets to explore, the RDF-distance and the stop-patterns to be considered (coherently with the the criteria for triples selection recalled above). Such settings tune the practical significance of the CS and, consequently, increase the effectiveness of communication.

We recall that r-graphs modeling CSs include blank nodes by construction (see Definition 1). To the best of our knowledge, no NLG tool is able to verbalize triples involving blank nodes in any position, so this is an original feature of our tool.

The tool works in three steps:

1. it takes as input a set of resources to be compared and application-specific parameters (datasets, RDF-distance, stop-patterns, uninformative triples);
2. it computes the CS parameterized as in Step 1
3. it generates a verbal explanation from the CS computed at Step 2

Step 3 implements a template-based approach, which allows developers to flexibly provide, as application-specific parameters, a dictionary for resources involved in triples, with particular reference to two kinds of undetermined resources: IRIs of involved blank nodes and IRIs of resources not further described in the datasets (even though not modeled as blank nodes).

5 Approach Demonstration

We show how we implemented our explanation approach with two use cases involving aggregation by similarity: the comparison of contracting processes in public procurement in the dataset released with TheyBuyForYou project [15] and the comparison and clustering of drugs in Drugbank [19]. In both cases, we show how to customize the approach w.r.t. the specific knowledge domain, to demonstrate its generality and flexibility.

5.1 Explaining Similarity of contracting processes in public procurement

The contracting process in procurement includes the procedures followed by a business entity when purchasing services or products: it starts when company managers identify a business need that must be fulfilled and ends when the contract is awarded and signed.

The proposal of easily accessible data-driven solutions supporting (especially public) contracting has been recently addressed. In particular, the Global Public Procurement Database (GPPD)[3] includes a country comparison functionality, that provides to users a side-by-side view of information on countries and regions, helping them to compare country profiles, procurement practices, laws and regulations, and performance indicators.

Also, a specific Contracting Data Comparison (CDC) platform[4] was developed by the Open Contracting Partnership (OCP), an independent non-profit organization born with the aim of publishing and using open, accessible and timely information on public contracting. To the best of our knowledge, none of such solutions is able to automatically compute implicit commonalities, to highlight shared features and explain the similarity of compared resources.

We here show our explanation approach w.r.t. the knowledge graph (in RDF) released with TheyBuyForYou project [15]. The project provides a platform with advanced knowledge-based tools for public procurement, including anomaly detection, cross-lingual document search, and storytelling tool. No tool for resources comparison and its explanation is provided with it.

Our method describes in English common language the commonalities among a set of contracting processes, by first computing their LCS.

TheyBuyForYou knowledge graph includes an ontology for procurement data, based on the Open Contracting Data Standard (OCDS) [16] . The OCDS data model is built around the concept of a contracting process, whose main phases are planning, tender, award, contract, and implementation.

In our example, we compare three RDF resources describing different contracting processes:

1. a public procurement issued by the Gateshead Council[5] for the supply of two lots of two 3.5t flatbed trucks with Tail-lift:
   `http://data.tbfy.eu/contractingProcess/ocds-0c46vo-0001-76e76119-992d-40ef-8444-7b020809ff81`
2. a tender for trucks supply issued by the Ringkøbing-Skjern municipality[6]:
   `http://data.tbfy.eu/contractingProcess/ocds-0c46vo-0133-026258-2019`
3. a public tender for the supply of trucks with multi-lift equipment and three containers issued by the district council of Azuaga[7]
   `http://data.tbfy.eu/contractingProcess/ocds-0c46vo-0203-2019-SUM-1`

[3] https://www.globalpublicprocurementdata.org/gppd/
[4] https://www.open-contracting.org/2014/04/30/comparing-contract-data-understanding-supply/
[5] https://www.gateshead.gov.uk
[6] https://www.rksk.dk
[7] http://www.azuaga.es

Recall from Sect.4 that the approach for computing an LCS in RDF asks for (1) the specification of the RDF-distance to be covered, (2) the list of patterns to ignore (stop-patterns) in the exploration, and (3) the list of triples to be removed in the final result because of their irrelevancy (uninformative triples).

In this example we set an RDF-distance equal to 2 for exploration, showing how a deep exploration of the knowledge graph may lead to significant similarity results. Thus, we first compute a significant CS at RDF-distance 2 of the three resources listed above, w.r.t. a set of stop-patterns and uninformative triples defined through the analysis of our examples. We show this CS in Figure 1. Then, we show in Figure 2 the natural language text generated from the CS in

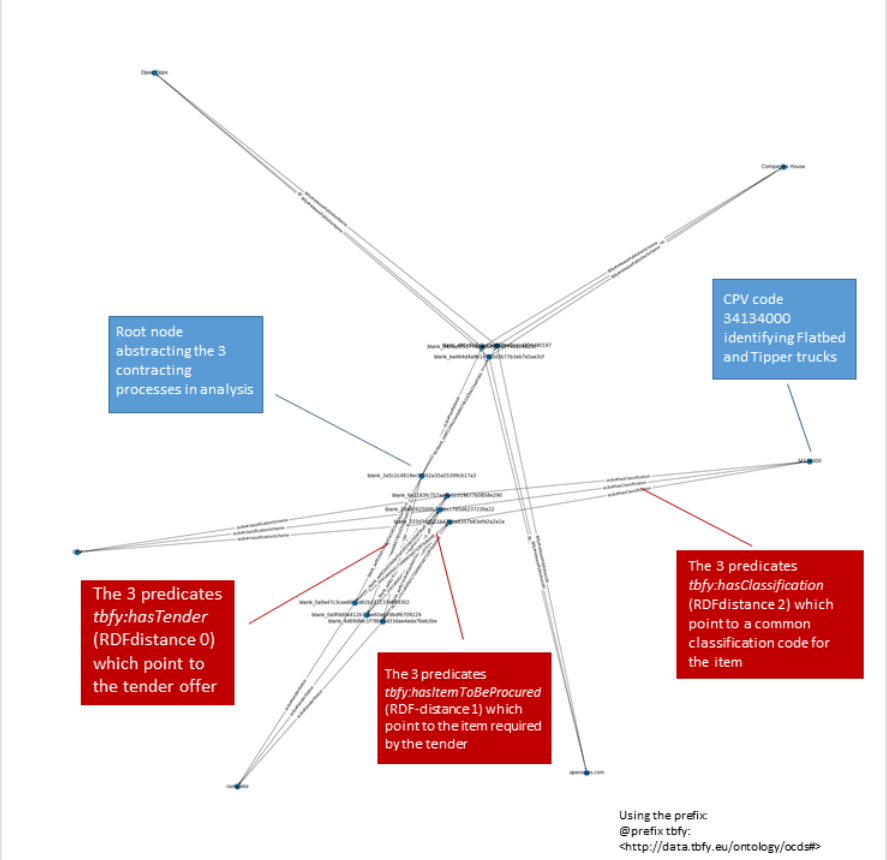

Fig. 1. A CS at RDF-distance 2 between three different contracting processes: i) a public tender for the supply of two flatbed trucks with tail-lift issued by the Gateshead Council; ii) a tender for the supply of trucks issued by the Ringkøbing-Skjern municipality; iii) a tender for the supply of trucks with multi-lift equipment and three containers issued by the district council of Azuaga

Figure 1. The reader may observe that the callouts in Figure 1 correspond to

```
The resources in analysis present the following properties in common:

1) They all have a release referencing some resource
 which  has publisher schema "Companies House"
 and  has publisher name "Open Opps"
 and  has publisher web page "https://openopps.com"

2) They all present a tender referencing some resource
 which  has tender status "complete"
 and  require a specific item(s) referencing some resource
  which  has classification schema "Common Procurement Vocabulary (CPV)"
  and  has classification code "34134000 (Flatbed and Tipper trucks)"
```

Fig. 2. Verbal explanation generated from the CS in Figure 1.

part of the content in description item 2) in Figure 2. In particular, the callouts show the full path (RDF-distance 2) from the CS root to the classification code "34134000".

5.2 Explaining Drugs Similarity

We here demonstrate the applicability of the proposed tool to the comparison of drugs. The need for evaluating the similarity of drugs emerges in several application scenarios, including drugs and/or side effects classification, search for substitute drugs and clustering, among others. In fact, several tools for drug comparison are available, in terms both of free services (see Drugs.com[8] and WebMD[9], among others) and of commercial solutions (*e.g.*, Lexicomp[10]). All such tools offer a parallel tabular view of explicitly declared drugs features, which ease the visual comparison of a small set of drugs selected by the user. None of them is able to extract implicit (logically deducible) commonalities and to highlight shared features.

The most advanced solutions addressing the analysis of drug similarity employ Machine Learning techniques based on metrics and return an aggregation of resources by similarity, without any universal and easy-to-read explanation. When dealing with resources that can be intrinsically represented by only numerical features, this lack of explanation is generally motivated by the complexity of the underling mathematical solving process. Values of the specific similarity measure represent the reason why resources are aggregated, as the only source of explanation.

Instead, when resources are described in formal languages endowed with semantics, like RDF, more informative and logic-based forms of explanation may

[8] urlhttps://www.drugs.com/compare/
[9] https://www.webmd.com/drugs/compare
[10] https://www.wolterskluwer.com/en/solutions/lexicomp/resources/facts-comparisons-user-academy/drug-comparisons

be returned. Our method describes in English common language the commonalities among a set of drugs defined in RDF, by first computing their LCS.

We first show such a human-readable explanation for the similarity of a pair of resources manually selected from Drugbank[11]: Amphetamine (`drugbank:DB00182`) and Phentermine (`drugbank:DB00191`), with the following prefix:

```
@prefix drugbank: <https://bio2rdf.org/drugbank> .
```

Our example refers to two values of RDF-distance, 0 and 1, to show the impact on final explanation of a deeper exploration of the information source. Stop-patterns and uninformative triples have been set according to heuristics evaluated in the performance of our examples.

Figure 3 shows a screenshot of our tool. The upper part includes the set of triples describing a significant CS of the analyzed pair of r-graphs when the RDF-distance of their triples is set to 0 (*i.e.*, only triples whose subject is the resource itself are included in the r-graph). Lower part of Figure 3 shows the explanation generated by the tool, starting from the triples in the upper part.

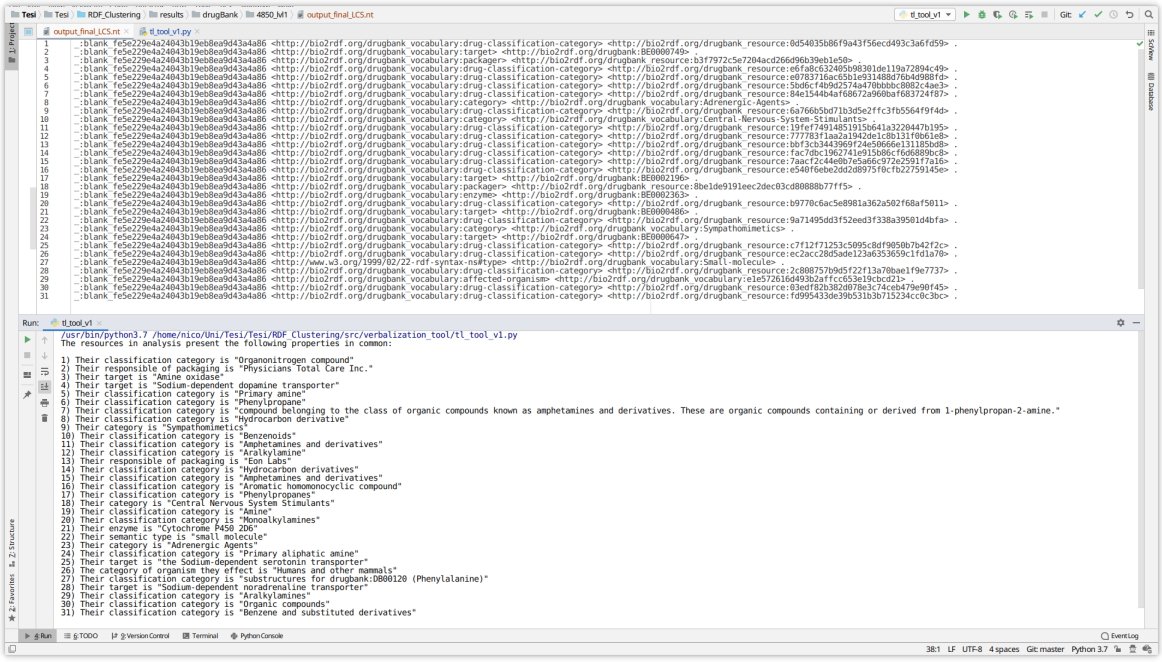

Fig. 3. A screenshot of the tool generating an explanation (lower window) for the RDF triples in a CS at RDF-distance 0 of a resources set (upper window).

Figure 4 shows a CS graph at RDF-distance 1 of the analyzed pair.

[11] https://old.datahub.io/dataset/fu-berlin-drugbank

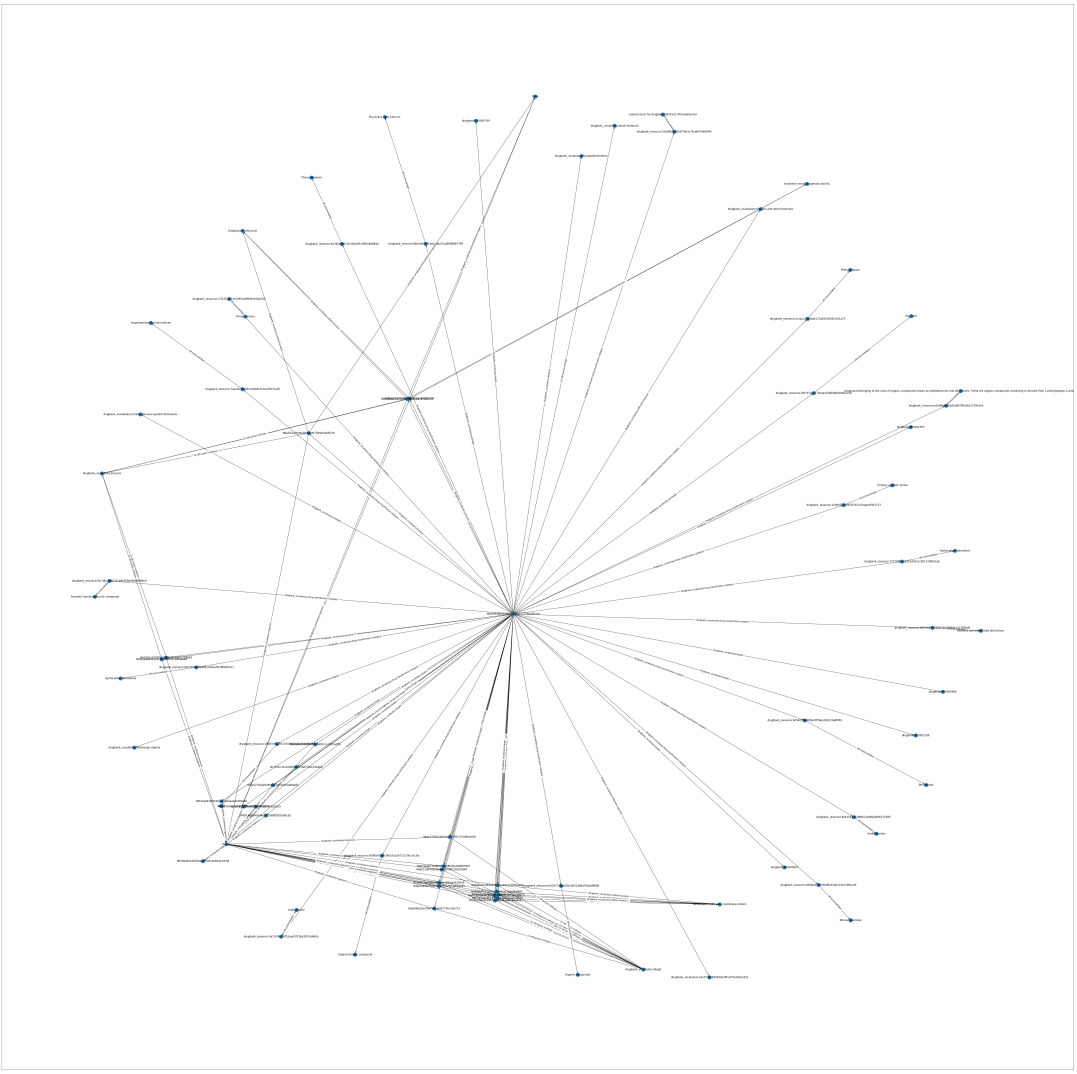

Fig. 4. A CS between Amphetamine (`drugbank:DB00182`) and Phentermine (`drugbank:DB00191`), computed from r-graphs including only triples at RDF-distance 1 from each resource). The figure is meant to give an idea of the complexity of the CS structure, without delving into details about involved resources. Note that this representation of the commonalities of the two resources, although pictorial, is still ineffective as an explanation.

In Figure 5, we show only the verbalization of triples added when passing from r-graphs with triples at RDF-distance 0 to r-graphs including also triples at RDF-distance 1. The complete explanation of the CS depicted in Figure 4 could be obtained by combining Figures 3 (lower part) and 5.

```
The resources in analysis present the following properties in common:

1) They share the property "target" each one of them referencing some resource
   which has organism type "Human"
   and  has cellular location "Membrane, multi-pass membrane protein"

2) They share the property "mechanism of effecting the body" each one of them referencing some resource
   which  has semantic type  "Mechanism of action"

3) They share the property "enzyme" each one of them referencing some resource
   which has organism type "Human"
   and has transmembrane regions of effect: "None"
   and has cellular location  "Endoplasmic reticulum"

6) They share the property "toxicity" each one of them referencing some  resource
   which  has semantic type "toxicity"
```

Fig. 5. Verbal explanation generated from the CS in Figure 4; we show only the verbal explanation of paths in the CS leading to triples at RDF-distance 1.

As the reader may observe, the number of triples collected in the CS at RDF-distance 1 is significantly high, but some of them may be considered uninformative, according to our analysis. In particular, in addition to the triples originally put in the set of uninformative triples (and then not returned in the graph in Figure 4), we excluded from the explanation also triples including URI not further described nor labeled in Drugbank.

We notice that our approach is able to generate a verbal explanation also from triples involving blank nodes, which are crucial in the computation of the LCS to abstract the commonalities of different resources at every RDF-distance form the root. To the best of our knowledge, such an ability is not available in any developed NLG tool. The informative potential of blank nodes treatment in our tool may be evaluated by looking at the content of Figure 5: all the sentences not in the first row of each figure item contain a relative sentence—starting with the relative pronoun "which"—that refers to "some resource" (an undetermined object represented by a blank node).

From now on, we show our explanation results, referring to a group of RDF resources aggregated by similarity thorough standard methods, with a twofold aim: *(i)* considering items supposed to be similar in an agnostic fashion, without manually selecting them; *(ii)* showing the behaviour of the explanation approach over groups of resources rather than pairs.

To this aim, we show the explanation of the commonalities shared by clusters of drugs returned by a standard clustering algorithm applied on Drugbank. In

particular, we applied the k-means algorithm in the Scikit-learn[12] Python library setting the number of clusters and the maximum number of iterations parameters equal to 125 and 400, respectively (such settings result from a standard validation process).

Figure 6 shows the explanation corresponding to the LCS (at RDF-distance 0) of all the 13 items returned in one of the clusters returned by the k-means implementation described above. The cluster included the following drugs:

```
Cathinone [drugbank:DB01560]            Aprindine [drugbank:DB01429]
Etidronic acid [drugbank:DB01077]       Metocurine [drugbank:DB01336]
Papaverine [drugbank:DB01113]           Methsuximide [drugbank:DB05246]
Ceftizoxime [drugbank:DB01332]          Pentosan Polysulfate [drugbank:DB00686]
Tipranavir [drugbank:DB00932]           Atomoxetine [drugbank:DB00289]
Clonazepam [drugbank:DB01068]           Paclitaxel [drugbank:DB01229]
Pentobarbital [drugbank:DB00312]
```

```
The resources in analysis present the following properties in common:

1) Their type is "small molecule"
2) They share the property "kingdom",
   all of them referencing the same resource not further described in the dataset
```

Fig. 6. Verbal explanation generated from the LCS at RDF-distance 0 of a cluster of RDF resources returned by k-means. Note that the common content has very little relevance for explanation.

Notably, the computed LCS is almost completely uninformative because it includes really generic features shared by items in the cluster. Apparently, the aggregation returned by k-means, that is purely numeric, does not reflect the logical content of analyzed resources. This behaviour suggests a possible byproduct of our tool: the produced explanation may be used as a fast-checker for the significance of results in the fine tuning of tools for clustering resources modeled in RDF.

6 Conclusion

We presented a logic-based methodology and a tool for the explanation of the similarities of a group of RDF resources, which can be pipelined to any Data-driven similarity aggregation tool, including Business Intelligence ones. Our methodology is agnostic w.r.t. the aggregation criterion and produces a verbal explanation of commonalities among groups of resources which the aggregation tool found to be similar.

Our tool works in two steps: first, the LCS of the analyzed set of resources is computed, as an abstraction of commonalities in the form of RDF triples; then,

[12] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

the LCS is translated into a verbal explanation, communicating only the effective knowledge. We presented the explanations given by our tool in two different use cases: Public Procurement and Drug comparison. Our methodology is domain-independent and can be adapted to several contexts (we omitted a third use case about Twitter accounts for lack of space). Also the significance of returned explanation may be tuned through the specification of patterns uninformative for the analyzed context.

Thus, our approach is potentially able to explain the similarity of RDF resources in every scenario, with a flexible level of communication effectiveness. The informative potential of our explanation is double-tied to the logic-based nature of the underlying theory, showing that the synergy between numerical and logical methods may improve explainability of Data-driven tools.

7 Acknowledgements

Projects Regione Lazio-DTC/"SanLo" (CUP F85F21001090003) and MISE (FSC 2014-2020)/ "BARIUM5G" (CUP D94I20000160002) partially supported this work.

References

1. Bouayad-Agha, N., Casamayor, G., Wanner, L.: Natural language generation in the context of the semantic web. Semantic Web 5(6), 493–513 (2014)
2. Cimiano, P., Lüker, J., Nagel, D., Unger, C.: Exploiting ontology lexica for generating natural language texts from RDF data. In: Proceedings of the 14th European Workshop on Natural Language Generation. pp. 10–19. Association for Computational Linguistics, Sofia, Bulgaria (Aug 2013), https://aclanthology.org/W13-2102
3. Colin, E., Gardent, C., M'rabet, Y., Narayan, S., Perez-Beltrachini, L.: The webNLG challenge: Generating text from DBpedia data. In: Proceedings of the 9th International Natural Language Generation conference. pp. 163–167 (2016)
4. Colucci, S., Donini, F., Giannini, S., Di Sciascio, E.: Defining and computing least common subsumers in RDF. Web Semantics: Science, Services and Agents on the World Wide Web 39, 62 – 80 (2016)
5. Colucci, S., Donini, F.M., Di Sciascio, E.: Common subsumers in RDF. In: Proc. of the 13th Conf. of Italian Assoc. for Artif. Intell. LNAI, vol. 8249. Springer (2013)
6. Colucci, S., Giannini, S., Donini, F.M., Di Sciascio, E.: A deductive approach to the identification and description of clusters in linked open data. In: Proc. of the 21st European Conf. on Artif. Intell. (ECAI 14). IOS Press (2014)
7. Ghosal, A., Nandy, A., Das, A.K., Goswami, S., Panday, M.: A short review on different clustering techniques and their applications. Emerging technology in modelling and graphics pp. 69–83 (2020)
8. Hayes, P., Patel-Schneider, P.F.: RDF 1.1 semantics, W3C recommendation (2014), http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/
9. Huang, L., Luo, H., Li, S., Wu, F.X., Wang, J.: Drug–drug similarity measure and its applications. Briefings in Bioinformatics 22(4) (11 2020)

10. Li, J., Zhang, Y., Qian, C., Ma, S., Zhang, G.: Research on recommendation and interaction strategies based on resource similarity in the manufacturing ecosystem. Advanced Engineering Informatics 46, 101183 (2020), https://www.sciencedirect.com/science/article/pii/S1474034620301543
11. Li, J., Cheng, G., Liu, Q., Zhang, W., Kharlamov, E., Gunaratna, K., Chen, H.: Neural entity summarization with joint encoding and weak supervision. In: Bessiere, C. (ed.) Proceedings of IJCAI-2020. pp. 1644–1650. ijcai.org (2020), https://doi.org/10.24963/ijcai.2020/228
12. Michalski, R.S.: Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts. Int. Journal of Policy Analysis and Information Systems 4, 219—-244 (1980)
13. Pérez-Suárez, A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: A review of conceptual clustering algorithms. Art. Intell. Review 52(2), 1267–1296 (2019)
14. Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O.P., Tiwari, A., Er, M.J., Ding, W., Lin, C.T.: A review of clustering techniques and developments. Neurocomputing 267, 664–681 (2017)
15. Soylu, A., Corcho, O., Elvesater, B., Badenes-Olmedo, C., Blount, T., Yedro Martinez, F., Kovacic, M., Posinkovic, M., Makgill, I., Taggart, C., Simperl, E., Lech, T.C., Roman, D.: TheyBuyForYou platform and knowledge graph: Expanding horizons in public procurement with open linked data. Semantic Web 13(2), 265–291 (2022)
16. Soylu, A., Elvesæter, B., Turk, P., Roman, D., Corcho, O., Simperl, E., Konstantinidis, G., Lech, T.C.: Towards an ontology for public procurement based on the open contracting data standard. p. 230–237. Springer-Verlag, Berlin, Heidelberg (2019), https://doi.org/10.1007/978-3-030-29374-1_19
17. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. p. 3104–3112. NIPS'14, MIT Press, Cambridge, MA, USA (2014)
18. Vougiouklis, P., Elsahar, H., Kaffee, L.A., Gravier, C., Laforest, F., Hare, J., Simperl, E.: Neural wikipedian: Generating textual summaries from knowledge base triples. Journal of Web Semantics 52-53, 1–15 (2018), https://www.sciencedirect.com/science/article/pii/S1570826818300313
19. Wishart, D.S., Knox, C., Guo, A.C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B., Hassanali, M.: Drugbank: a knowledgebase for drugs, drug actions and drug targets. Nucleic acids research 36(suppl 1), D901–D906 (2008)
20. Yu, Y., Umashankar, N., Rao, V.R.: Choosing the right target: Relative preferences for resource similarity and complementarity in acquisition choice. Strategic Management Journal 37(8), 1808–1825 (2016), https://onlinelibrary.wiley.com/doi/abs/10.1002/smj.2416
21. Zhou, G., Lampouras, G.: WebNLG challenge 2020: Language agnostic delexicalisation for multilingual RDF-to-text generation. In: Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+). pp. 186–191. Association for Computational Linguistics, Dublin, Ireland (Virtual) (12 2020), https://aclanthology.org/2020.webnlg-1.22