Improving KG Completeness Evaluation considering Information Need as First-Class Citizen

Jennie Andersen, Sylvie Cazalens, Philippe Lamarre

Improving KG Completeness Evaluation considering Information Need as First-Class Citizen

ANR Project DeKaloG*
Research report D3.2

Jennie Andersen, Sylvie Cazalens, and Philippe Lamarre
Univ Lyon, INSA Lyon, CNRS, UCBL, LIRIS, UMR5205,
F-69621 Villeurbanne, France

December 5, 2022

1 Introduction

Knowledge Graphs (KGs) are at the core of numerous tasks, such as developing applications or services, verifying of information, answering a federated query, etc. When choosing or using a KG, it is essential to easily characterize its specificities, strengths, and weaknesses. Especially, there is an increasing demand for transparency as well as accountability. Transparency may have many definitions [26] but it is mainly concerned with the accessibility of information as well as its presence. For its part, accountability has the main objective of verifying the presence of fairly accurate information about people's responsibilities and actions [16]. Whatever their differences, information need is the main component of both concepts. It is important to notice that different people may have different expectations, therefore, the need of information is contextual. More generally, we want to cover all kinds of information needs, especially those which are fine-grained. This could be useful for anyone who is willing to evaluate a Knowledge Graph based on its information, whether it concerns transparency or any other context.

Among the existing tools and methods to evaluate KGs, such as for measuring data quality [28] or FAIRness (FAIR: Findability, Accessibility, Interoperability, Reusability) [23], only a few measures focus specifically on information within the KGs. Those who do concern information widely consider as useful, looking for a creator or a license for instance. For information which is more contextual, the closest are measures of completeness. However, from our point of view, completeness in the literature suffers two weaknesses: firstly, the need of information is not sufficiently explicit. Secondly, existing completeness measures too often rely on query that are limited in there form (only a few triples) and thus in what they enable to obtain.

In this paper, we seek to fill this gap by proposing a method to evaluate the completeness of a KG regarding an explicit list of information defined as necessary. Therefore, we focus on the notion of *information need*, enabling to formally describe and manipulate the required information. Hence, we aim at answering the research question: "How to evaluate the ability of a KG to answer an information need?" with the underlying question "What is an information

*AAPG2019 - ANR Project - Funding Instrument PRC, CE3 - Intelligence Artificielle.

need and how to define it?". Our method can therefore be applied to concepts relying on the presence of information such as transparency and accountability. The second objective of this paper is to provide a first information need, focused on accountability, for which a set of questions of reference has been defined [16]. The proposal is then used to evaluate the accountability of existing knowledge graphs (more precisely RDF graphs), which is also a partial evaluation of their transparency. This starts with the description of the list of questions to be answered, but also the structure to organize them and their relative importance. Then, questions are translated into queries in order to evaluate the KG. All of this constitute the *need*. The evaluation of completeness regarding the need results in a global and some local scores of completeness. The structuring elements allows to navigate on and explore these results, and also to identify the weaknesses of the KG. All of this allows any user to understand how the scores were produced, to check the answers provided by the KG, and the whole process of building the need. It also helps to identify areas of improvement.

The definition of a need provides a frame to explicitly define what is required to meet even vague concepts such as transparency. Therefore, the evaluation of completeness regarding an information need may be useful to all users of knowledge graphs, from data producer to data consumer, as well as for producer of KG indexes or people and organization providing recommendation (FAIR...) on semantic web. It also applies at different scales, for a prototypical need shared by most of the people, or for a very specific and individual need. We aim at using it for the construction of an index of KGs by explicitly building prototypical needs, starting with one about transparency. The results may be used to evaluate, classify, select some of the indexed KGs according to these needs.

Paper organization In the first section, we present existing works on possible metrics (data quality, FAIRness) to audit a dataset. We also identify some needs of verifications expressed on transparency. Then, we present our proposal to formalize information needs in the section 3. It presents how to represent and manipulate an information need and its structure and it defines the associated completeness of a KG. Next, the section 4 is dedicated to the application of our proposal on the questions introduced by LIQUID [16] on accountability. We confront several KGs to this need and evaluate their completeness. Therefore, it provides a first measure of accountability of KGs. Finally, we discuss several points of the presented approach.

2 State of the art

The following state of the art complement the one of our previous report on transparency [4] which details various definitions of transparency according to the domains and contexts. In this section, we remind what transparency is and its relation with accountability before focusing on the evaluation of knowledge graphs.

2.1 Transparency and accountability

Although transparency is cited in many papers and in various domains, it remains a rather ill-defined concept when going beyond the general definition of a dictionary. There is no uniform view of what constitutes it [15], even when only computer sciences are considered.

The various definitions of transparency mostly require the access to more information. The main problem is therefore to define what information to disclose. It can be asked for monitoring purposes: "The availability of information about an actor that allows other actors to monitor the workings or performance of the first actor." [19]. Or it can be related with privacy and personal

data: "Data transparency is the ability of subjects to effectively gain access to all information related to data used in processes and decisions that affect the subjects." [6] ; or "to verify which aspects of the data determine its results." [11]. These selected examples show that the definitions of transparency are very contextual. Obviously, they depend on the subject to be transparent. But even for the same object, there can be significant variations, as shown by Bertino [6].

Moreover, transparency can only be truly attested by an external observer (subjects, the public...). Hence, transparency requires an observed element, an observer [19], and a mean of observation [26]. Finally, transparency is needed for a specific purpose which is user defined.

Hence, in order to apply to any context, we propose the following definition of transparency: *Transparency aims at giving access to the suitable information regarding the needs of information to any observer.* So, to obtain transparency, it is mandatory to provide, to give access to information suitable regarding the need. It arises as a question of completeness between the information needed to meet the need and the information actually available in the dataset.

As far as we know, there does not exist any measure of dataset's transparency. In order to find some requirements of transparency, we may consider accountability and the LiQuID metadata model, provided by Oppold and Herschel [16]. Indeed, transparency is often related with accountability [22] and is sometimes even considered as synonymous [15]. Accountability requires transparency and goes beyond the general need of more information: it specifically requires information involving people's responsibility and justification or verification of data use or misuse. Dataset accountability means that "there is sufficient information to justify and explain the actions on [the] datasets to a forum of persons, in addition to descriptive information and information on the people responsible for it" [16].

The LiQuID metadata model is defined to make datasets accountable throughout their life-cycle. Therefore, it provides a list of requirements that must be met to be consider accountable and that can be used as a basis for a measure.

2.2 Evaluating Knowledge Graphs

There are several ways of auditing and measuring KGs among which one can find the data quality, the FAIR principles, and several tools of monitoring. They rely on different concepts and metrics which can be classified into three categories presented thereafter.

First, data quality provides an extensive list of metrics covering a lot of concerns. It is "commonly conceived as fitness for use for a certain application or use case" [28]. Two points seem important in this formulation: the notion of "fitness for use" and the fact that this notion is to be considered from the point of view of a particular application or use case. A large volume of the quality literature has been devoted to specifying different facets of "fitness for use": availability, accuracy, conciseness, completeness, understandability, timeliness, etc. And, most of the time, the work is focused on the study of general elements that can be beneficial to any application.

To organize all these characteristics, Wang et Strong [21] introduce a framework for assessing the data quality, then it has been adapted for knowledge graphs [28, 10, 9]. In this framework, data quality is divided into several "categories". Each category is sub-divided into "dimensions" that contains one or several "criteria". Finally, each data quality criterion is associated with a metric in order to measure it on a given knowledge graph. The most common data quality categories are the following [21, 10, 9]. Intrinsic category "denotes that data have quality in their own right" [21] and is therefore "independent of the user's context" [28]. The contextual category "highlights the requirement that data quality must be considered within the context of the task at hand" [21]. Then, the representational category evaluates "how well the data is represented in terms of common best practices and guidelines" [9]. Finally, accessibility category

covers "aspects related to the access and retrieval of data to obtain either the entire or some portion of the data for a particular use case" [28]. Zaveri et al. add two new categories, the trust category which focuses "on the perceived trustworthiness of the dataset" and the dataset dynamicity category which covers dataset's "freshness over time, the frequency of change over time and its freshness over time for a specific task" [28].

Both data quality and transparency condition their concept to a specific context, data quality asks datasets to fit the use regarding a context while transparency requires information provided by a dataset to be suitable regarding a context. They aim at satisfying a need, but for data quality, this need is supposed to be shared by the majority while for transparency it could be very specific. More specifically, transparency undoubtedly fits the definition of the contextual category of data quality and could therefore be considered as part of this category.

Some aspects of data quality can also be found among the FAIR guiding principles [23]. Instead of defining metrics, they provide precise guidelines to help producers improving the *Findability*, *Accessibility*, *Interoperability*, and *Reusability* of their dataset in order to "enhance the reusability of their data holdings" [23]. FAIR is not specific to KG, it aims at applying to any dataset. Different implementations were developed either to help providers increase the FAIRness of their data [18] or to evaluate the FAIRness of a dataset [17, 2, 3]. Some metrics were also defined by the original authors to assess how well a KG complies with these guidelines [24]. While FAIR principles obviously contribute to the quality of a dataset through the accessibility, interoperability, and some aspects of reusability, they also bring a new insight with the findability.

Finally, there are some catalogs of KGs that provide more information about them through metrics. SPARQLES [20] measures discoverability, interoperability, performance, and availability. YummyData [27], specialized in biomedical KGs, aggregates several metrics of quality into what is called the Umaka score. It relies on six dimensions : availability, freshness, operation, usefulness, validity, and performance. They both use metric closely related with existing dimensions of data quality or FAIRness.

All these auditing tools provide metrics or guidelines that inform about the quality of the data. FAIR and monitoring tools add some precisions to data quality and even new dimensions. Among all these tools, three different types of metrics seem to appear. The first one is independent of the data and concern the system that hosts the RDF dataset. Then, metrics about the form of the data concerns metrics that are not interested in the content itself or the meaning of the data, but on how it is written. Finally, some metrics focus on the content, on the information conveyed by the data. It is important to notice that these three types are not disjoint. For instance, vocabularies both concern the content because they provide information on the domain covered by the KG and the form of the data because they give insight into the schema and structure of the KG.

2.2.1 Metrics concerning the hosting system

The first type of metrics focuses on the system that hosts the dataset and makes it available. Hence, it mostly concerns accessibility, both in data quality and FAIR. The latter is mainly concerned by the protocol and condition of access to the dataset while the former brings some precisions specific to knowledge graphs: for instance, a SPARQL endpoint should be available as well as a dump. Data quality also provides some metrics of performance of the dataset service (latency, throughput, scalability) [28].

In FAIR, findability includes other principles relying on that system, including that data must be "registered or indexed in a searchable resource" [23]. Indeed, many measures of FAIRness only rely on sending HTTP requests [24] that do not care about the data.

Most of the measures done by SPARQLES and YummyData also focus on the hosting system.

Indeed, they do not care about the data of the KG for availability, performance, interoperability, and operation as well as one of the two metrics of validity. This can be explained by the fact that they aim at informing about the usability of a list of KGs, which start by these technical considerations.

2.2.2 Metrics concerning the form of the data

Metrics about the form of the data are not interested in the content itself or the meaning of the data, but on how it is written: the syntax, the schema used, the respect of rules, but also the number of triples, properties, etc.

Consistency, conciseness, amount of data and many other dimensions are associated with metrics concerning the form of the data. For instance, they rely on computing the number of triples that do not respect some RDF inference rules or OWL rules, checking the use of some vocabularies and some properties, counting the number of unique objects, etc. This results in a majority of statistical measures about the dataset. Indeed, in the survey by Zaveri et al. [28], "most of the metrics take the form of a ratio, which measures the occurrence of observed instances out of the occurrence of the desired instances". In FAIR, these considerations are supported by the interoperability which first requires a "formal, accessible, shared, and broadly applicable language for knowledge representation" and the use of "vocabularies that follow FAIR principles" [23].

While SPARQLES has no interest in the form of the data, it appears in two aspects of YummyData: operations where the number of properties, labels, classes and datatypes are counted and one metric of validity.

2.2.3 Metrics concerning the information conveyed by the data

The last type of metrics concerns the information conveyed by the data, by requiring specific information within the dataset.

Indeed, some metrics do require specific information, conveyed by data or metadata. For instance, simple elements such as the "presence of the title, content and URI of the dataset" are mentioned in data quality studies [28]. Also, the presence of a license is a shared requirement. The reusability component of the FAIR principles asks that "(meta)data are released with a clear and accessible data usage license" [23]. It is also a dimension of the accessibility category of data quality [28]. Similarly, FAIR principles and data quality studies both require provenance information. It can be seen as part of reusability (FAIR principles) or more specific, as for example, the provenance dimension introduced within the contextual category in [9].

All these metrics are very specific to given information considered as very essential. But in some cases, the information required is not as consensual as the need of a license or depends more on the context or the domain. In the contextual category of the data quality studies, the completeness dimension fills this gap. It "refers to the degree to which all required information is present in a particular dataset" for a given task [28]. As defined in [10], it is divided into three metrics: schema completeness, property completeness (or column completeness), and population completeness. Population completeness (respectively schema completeness) uses a gold standard which defines the entities (respectively, the classes and properties) that should be represented in the KG. Column completeness evaluates if for all entities of a given class, there exists a value for a given property.

A survey focused on Knowledge Graphs Completeness [12] identifies four other types of completeness: interlinking completeness, currency completeness, labelling completeness and metadata completeness. The first three ones are mainly assessed by statistical metrics, measuring

respectively how much instances are interlinked with other KGs, the availability of valid elements over different periods of time, and the presence of labels. Metadata completeness is defined by "the degree to which metadata properties and values are not missing in a dataset for a given task" [12]. Because transparency often requires meta-data, it is part of metadata completeness.

However, more generally, the existing metrics are not sufficient to evaluate KG completeness regarding information defined as necessary. Indeed, these metrics are partial and do not allow to verify the presence of a multitude of diverse information. We do not limit ourselves to schema or values of a specific property, we would like to use any query, without limitations. Moreover, some of them require a gold standard to be provided, whereas we suppose that information is not known beforehand, only the kind of information wanted (e.g. the authors). So, we aim at improving the completeness by allowing users to define the needed information (with more liberty than a gold standard) and resulting in a measure of completeness regarding the need of the user, whatever need of information it is. Therefore, our measure of completeness is transversal among the existing types of completeness.

3 Formalizing information need

In this section, our goal is to define completeness of a Knowledge Graph (KG) w.r.t. some information need. Instead of defining an information need as a simple set of questions, we introduce the notion of analysis dimension to structure it using a rooted tree of tags. This tree is a way to reflect a progressive and systematic way to identify and classify the questions that seem relevant. Then, an information need consists in tagging questions with tags of an analysis dimension. We introduce completeness as an aggregation of question evaluations independently of any implementation language. After evaluation, the tree structure enables navigating across the results. We then discuss implementation issues.

3.1 Analysis Dimension

We aim at defining a simple way to structure the questions in a progressive and systematic way, just as one can do when analyzing a problem. This echos to methods such as the well-known 5Ws[1], or the Goal-Question-Metric approach [5]. For example, one could use different domains and then the 5W question types. To our view, a weighted tree structure of tags is both generic and simple enough to clearly and simply specify how to organize a set of questions. Weights are in addition used to indicate the relative importance of the tags. They will impact the evaluation of completeness.

3.1.1 Formal definition

Definition 3.1 (Analysis dimension)**.**
*An **Analysis Dimension** Δ is a weighted rooted tree $\Delta = \langle T_\Delta, R_\Delta, \delta_\Delta \rangle$ such that*

- *T_Δ is a set of tags, i.e. the set of nodes of the tree.*
- *$\top \in T_\Delta$ is the root of the tree.*
- *R_Δ is a relation structuring tags and defining the tree: $R_\Delta \subset T_\Delta^2$. Let $t, t' \in T_\Delta$, then $tR_\Delta t'$ means that t is the parent node of t'.*

[1] `https://en.wikipedia.org/wiki/Five_Ws`

- δ_Δ: $T_\Delta \to \mathbb{R}^{+*}$ *is a function giving the weight of a tag. The weight is relative to the siblings of the node in the rooted tree.*
  Let $t, t_1, t_2 \in T_\Delta$, *such that* $tR_\Delta t_1$ *and* $tR_\Delta t_2$, *then,* $\delta_\Delta(t_1) = \alpha\delta_\Delta(t_2)$ *means* t_1 *is* α *times more important than* t_2 *in the context of* t.

Notice that, since, by construction $\top$ has no brother, its weight is of no interest. It will therefore never be indicated in the following.

An analysis dimension is flat when there is no hierarchy between tags except with $\top$. They are simple and easy to define and use.

Property 3.1 (Flat analysis dimension)**.**
An analysis dimension is said to be flat iff the height of the analysis dimension tree is 1.

Example 3.1 (Six Ws questions as an analysis dimension)**.**
To obtain an analysis dimension corresponding to the Ws questions (or WH-questions), one can use the following flat analysis dimension. $\Delta_{6W} = \langle T_{\Delta_{6W}}, R_{\Delta_{6W}}, \delta_{\Delta_{6W}} \rangle$ *such that:*

- $T_{\Delta_{6W}} = \{\mathit{Who}, \mathit{What}, \mathit{When}, \mathit{Where}, \mathit{Why}, \mathit{How}\}$
- $R_{\Delta_{6W}} =$
  $\{\top R_{\Delta_{6W}} \mathit{Who}, \top R_{\Delta_{6W}} \mathit{What}, \top R_{\Delta_{6W}} \mathit{When}, \top R_{\Delta_{6W}} \mathit{Where}, \top R_{\Delta_{6W}} \mathit{Why}, \top R_{\Delta_{6W}} \mathit{How}\}$
- $\delta_{\Delta_{6W}}(\mathit{Who}) = 1$; $\delta_{\Delta_{6W}}(\mathit{What}) = 1$; $\delta_{\Delta_{6W}}(\mathit{When}) = 1$; $\delta_{\Delta_{6W}}(\mathit{Where}) = 1$; $\delta_{\Delta_{6W}}(\mathit{Why}) = 1$; $\delta_{\Delta_{6W}}(\mathit{How}) = 1$.

Here, all questions are of equal importance and do not depend on each other. Figure 1 presents this analysis dimension as a rooted tree. Weights of tags are indicated on the edge linking them to their parent.

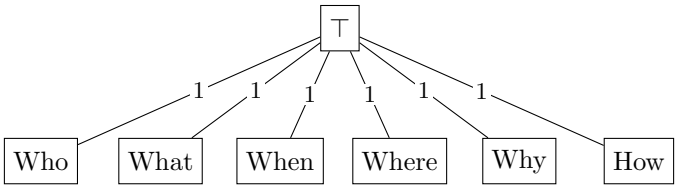


Figure 1: 6Ws Analysis Dimension

Example 3.2 (A hierarchy to analyze a photograph)**.**
It is also possible to have more level in the hierarchy of tags. For instance, if some want to analyze a photograph, they can use the following hierarchy in order to organize their questioning. The analysis dimension Δ_P *is composed of two levels. The first level of tags concerns the three main parts of a picture analysis: the identity of the photograph (title, year, author...), its description, and its interpretation.*

- *id.: The identity of the photograph concern its title, author, date, localization...*
- *desc.: The description of the visual content of the photograph.*
- *int.: The interpretation one makes of the photograph.*

All theses tags can be divided into other tags. For instance, the identity of the photograph either concerns the characteristics of the author or the photograph itself.

- *author: Identity of the photographer (name, date of birth, ...)*
- *photograph: Information to identify the photograph, including its title, where and when it was taken.*

Figure 2 presents the corresponding analysis dimension as a tree. Weights have been added arbitrarily: the interpretation tag is more important than the identity tag which is itself more important than the description tag. Regarding the identity, the two children are of same importance, while regarding Description, two of the tags are twice more important than the last one.

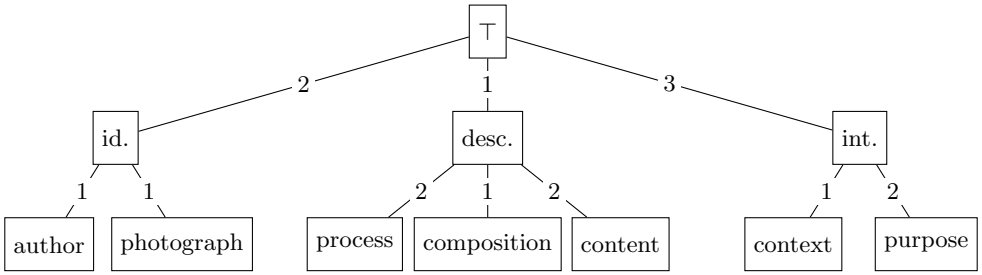


Figure 2: Photograph Analysis Dimension

3.1.2 Operators on analysis dimensions

We introduce two operators to manipulate analysis dimensions. The resulting simple algebra is enough to easily create a dimension from others and, in a first step, we do not feel the need to enrich it with other operators.

First, we introduce an operator that enables focusing on some tags only. It is not necessary to list all the tags to be kept: indicating those which are the most precise (the deepest in the tree) is enough.

Definition 3.2 (Restriction)**.**
Let $\Delta = \langle T_\Delta, R_\Delta, \delta_\Delta \rangle$ be an Analysis Dimension. Let S be a non empty set such that, $S \subseteq T_\Delta$. A ***restriction*** *of Δ w.r.t. S, noted $\Delta|_S = \langle T_{\Delta|_S}, R_{\Delta|_S}, \delta_{\Delta|_S} \rangle$ which is the smallest weighted subtree of Δ, rooted at $\top$, such that $S \subseteq T_{\Delta|_S}$. More formally:*

- $T_{\Delta|_S} = S \cup \{t \in T_\Delta \mid \exists t' \in S, t \text{ is an ancestor of } t' \text{ in } \Delta\}$
- $R_{\Delta|_S} = R_\Delta \cap T^2_{\Delta|_S}$
- $\delta_{\Delta|_S} = \delta_\Delta|_{T_{\Delta|_S}}$

Theorem 3.1.
Let $\Delta = \langle T_\Delta, R_\Delta, \delta_\Delta \rangle$ be an Analysis Dimension. Let S be a non empty set such that, $S \subseteq T_\Delta$. $\Delta|_S$ is an analysis dimension.

Proof.

- Since $S \subseteq T_\Delta$, then by definition $T_{\Delta|_S} \subseteq T_\Delta$. As T_Δ is a set of tags, then $T_{\Delta|_S}$ is also a set of tags.
- By hypothesis, S is a non empty subset of T_Δ. Either $S = \{\top\}$ then by definition $S \subseteq T_{\Delta|_S}$ and $\top \in T_{\Delta|_S}$. Or, $\exists t \in S$ such that $t \in T_\Delta \setminus \{\top\}$, therefore $\top$ is an ancestor of t in Δ and by construction, $\top \in T_{\Delta|_S}$.

- Let us show that $R_{\Delta|_S}$ defines a tree rooted at $\top$. First, $R_{\Delta|_S} \subseteq R_\Delta$, and as R_Δ defines a rooted tree, it is acyclic. So $R_{\Delta|_S}$ is also acyclic. Then, as shown before, $\top \in T_{\Delta|_S}$. Finally, let us show that $R_{\Delta|_S}$ is connected. Let $t \in T_{\Delta|_S} \setminus \top$. If $t \in S$, then all its ancestors in Δ belong to $T_{\Delta|_S}$. So by construction of $R_{\Delta|_S}$, there exists a path from t to $\top$. Else, if $t \notin S$, then $\exists t' \in S$ such that t is an ancestor of t'. As all ancestors of t are ancestors of t', it means that all ancestors of t in Δ belong to $T_{\Delta|_S}$. Then by construction of $R_{\Delta|_S}$, there exists a path from t to $\top$. Therefore, $R_{\Delta|_S}$ defines a tree rooted at $\top$.
- $\delta_{\Delta|_S}$ is clearly a function giving weight to each tag of $T_{\Delta|_S}$.

□

Example 3.3 (Restricting the six Ws questions)**.**
The five Ws analysis dimension can be defined as $\Delta_{5W} = \Delta_{6W}|_{\{Who, What, When, Where, Why\}}$.

Example 3.4 (Restricting the photograph analysis dimension)**.**
A more complex example corresponding to $\Delta_P|_{\{id., desc., process, content, purpose\}}$ *is presented Figure 3. Every leaf tag which does not appear in the restriction set is removed from the tree. A parent tag is removed from the tree only if it does not appear in the restriction set and if all its children were removed. Hence, even if it does not appear in the restriction set, "int." is still in the tree. Indeed, as its child "purpose" has been kept, the resulting tree must include the original edge between "purpose" and "int." to be a subtree of the original analysis dimension.*

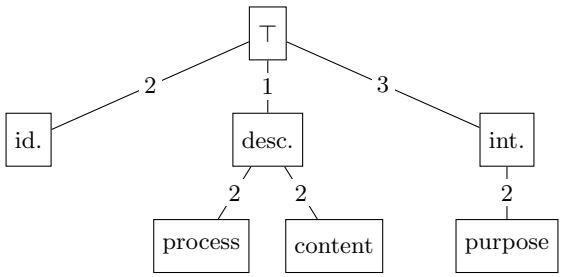


Figure 3: Restriction of the Photograph Analysis Dimension

The second proposed operator aims to combine two analysis dimensions in order to build complex dimensions in a modular way, for example starting from a flat dimension and then *extending* it with another and so on.

Definition 3.3 (Extension)**.**
Let Δ_1 *and* Δ_2 *be two Analysis Dimensions. The* ***extension*** *of* Δ_1 *with* Δ_2*, noted* $\Delta_1 \lhd \Delta_2$ *is defined by:*

- Δ_1 *is a subtree of* $\Delta_1 \lhd \Delta_2$.
- *For each leaf* t_1 *of* Δ_1*, the corresponding node* t *in* $\Delta_1 \lhd \Delta_2$ *is the root of a subtree corresponding to* Δ_2 *where* t *takes the place of* $\top$.

Notation: Whenever it is necessary to distinguish the different instances of Δ_2, a dotted notation will be used. For example, $t_1.t_2$ denotes the node t_2 of Δ_2 which is copied into Δ to create the subtree of t_1.

Theorem 3.2.
Let $\Delta_1 = \langle T_{\Delta_1}, R_{\Delta_1}, \delta_{\Delta_1} \rangle$ *and* $\Delta_2 = \langle T_{\Delta_2}, R_{\Delta_2}, \delta_{\Delta_2} \rangle$ *be two Analysis Dimensions. Then* $\Delta_1 \lhd \Delta_2$ *is an Analysis Dimension.*

Proof.

- Δ_1 is a subtree of $\Delta_1 \lhd \Delta_2$. Hence, $T_{\Delta_1} \subseteq T_{\Delta_1 \lhd \Delta_2}$ and $T_{\Delta_1 \lhd \Delta_2}$ is a non empty set of tags.
- Δ_1 is an analysis dimension, so $\top \in T_{\Delta_1}$. Since $T_{\Delta_1} \subseteq T_{\Delta_1 \lhd \Delta_2}$, then $\top \in T_{\Delta_1 \lhd \Delta_2}$.
- $R_{\Delta_1 \lhd \Delta_2}$ defines a tree rooted at $\top$. As $\top \in T_{\Delta_1 \lhd \Delta_2}$, let us show that each tag except the root has a unique parent tag according to $R_{\Delta_1 \lhd \Delta_2}$ and that the root has no parent node. Let $t \in T_{\Delta_1 \lhd \Delta_2} \setminus \{\top\}$. Either $t \in T_{\Delta_1} \setminus \{\top\}$. Then, t belongs to Δ_1, the upper part of the tree, and t has a unique parent in Δ_1 and therefore in $\Delta_1 \lhd \Delta_2$.
  Or t belongs to a copy of Δ_2, so $t = t_1.t_2$ with $t_2 \in T_{\Delta_2} \setminus \{\top\}$. As Δ_2 is an analysis dimension, t_2 has a unique parent t_2' in Δ_2. Then, t also have a unique parent in $\Delta_1 \lhd \Delta_2$ which is either $t_1.t_2'$ if $t_2' \neq \top$, or t_1 if $t_2' = \top$.
- As $R_{\Delta_1 \lhd \Delta_2}$ defines a rooted tree, then the weights on each node of the tree are preserved from Δ_1 and Δ_2 to their copies in $\Delta_1 \lhd \Delta_2$. Indeed, δ_Δ is defined as follows

$$\delta_\Delta(t) = \begin{cases} \delta_{\Delta_1}(t) & \text{if } t \in T_{\Delta_1} \\ \delta_{\Delta_2}(t_2) & \text{if } t = t_1.t_2 \text{ with } t_1 \in T_{\Delta_1}, t_2 \in T_{\Delta_2} \end{cases}$$

  which is a function giving weight to each tag of $T_{\Delta_1 \lhd \Delta_2}$.

□

One must be careful when using this extension operator in order for the resulting dimension to make full sense. Indeed, some tags of the extending dimension could be semantically not compatible with some of the extended one. If in a particular context, a tag makes no sense at all, it would be much better not to keep it. Even if this extreme situation is not reached, special attention should be paid to the weight of the tags. Indeed, the relative importance of the tags of T_{Δ_2} which are copied in $\Delta_1 \lhd \Delta_2$ may semantically vary depending on the different contexts defined in Δ_1. Interestingly, those two problems are avoided if the two dimensions are independent.

Definition 3.4 (Independent dimensions)**.**
Two dimensions are independent if classifying an element with respect to one dimension does not give any information about the classification of the same element in the other dimension.

In summary, it is advisable to use the extension operator only when the two dimensions are independent. Such verification is matter of semantic considerations that are beyond the scope of this paper. It is therefore up to the designer to consciously choose whether or not to use the operator, even if she considers that the dimensions are not independent. Of course, she is also free to make any modifications to the resulting analysis dimension to ensure its semantic consistency (for example, by restricting it).

Example 3.5 (Extending the photograph analysis dimension with 5Ws)**.**
Let us consider only the first level, Δ_{P1}*, of the Photograph Analysis Dimension. It is possible to extend it with the 5Ws Analysis Dimension. Hence, Figure 4 illustrates* $\Delta_{P1} \lhd \Delta_{5W}$*.*

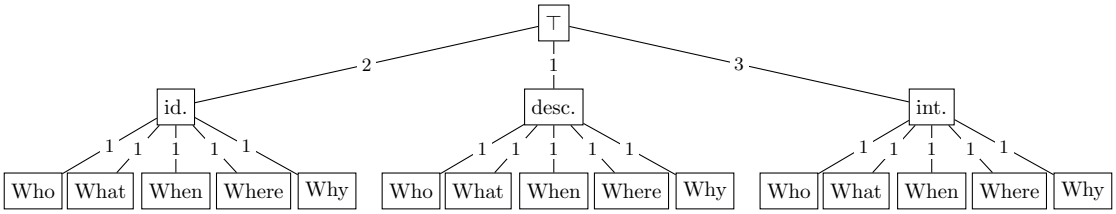


Figure 4: Extension of the first level of Photograph Analysis Dimension with the 5Ws one

It is frequent that a problem, an object, must be analyzed from different points of view. It is therefore often necessary to use a set of analysis dimensions. Insofar as the introduced operators make it possible to produce new dimensions of analysis from existing ones, it becomes interesting to look for the minimal set allowing to generate the desired set. Rather than constructing each analysis dimension one by one, only the minimal set has to be constructed, the other being generated using the introduced operators. Thanks to this operator, several working methodologies are open to analysis dimension designers. One may choose to build a complex dimension directly, while another may prefer to work on several simple and independent ones which she will combine.

3.2 Information Need

In this section, we formally define information needs and associated operators.

3.2.1 Formal definition

An information need links a set of questions to the tags of an analysis dimension. In addition, we consider weights to express the relative importance of a question compared to those that are linked to the same tag. Then the tagging can be seen as a set of triples, each one associating a question to some tag with some weight.

Definition 3.5 (Information need)**.**
*An **Information Need**, noted $\mathcal{N}$, is a tuple $\mathcal{N} = \langle \mathcal{Q}, \Delta, Tag_{\mathcal{Q},\Delta} \rangle$ such that*

- *$\mathcal{Q}$ is a non empty set of questions.*
- *Δ is an analysis dimension.*
- *$Tag_{\mathcal{Q},\Delta}$ is a tagging, meaning a set : $Tag_{\mathcal{Q},\Delta} \subset \mathcal{Q} \times T_\Delta \times \mathbb{R}^{+*}$ where $(q, t, \omega) \in Tag_{\mathcal{Q},\Delta}$ denotes that q is tagged with t and with a weight ω relatively to other elements of $\mathcal{Q}$ also tagged with t.*

Some information needs have interesting characteristics. For example, it seems more reasonable that all the questions present in the need are related in some identified way to the analysis dimension, i.e. that there are no irrelevant questions. It seems equally relevant that the questions cover all aspects of the analysis dimension; i.e. that no part of the analysis dimension is left without any question. Moreover, to hope to obtain interesting answers, it is important that the questions are as precise as possible, and so that they are expressed in the most precise context that an analysis dimension can provide, i.e. one of its leaves. All these considerations are grouped together in the notion of a well-formed need.

Definition 3.6 (Well-formed information need)**.**
*A **well-formed information need** is an information need which satisfies the following properties:*

- *Unicity of weights:* $\forall(q,t,\omega_1,\omega_2) \in \mathcal{Q} \times T_\Delta \times \mathbb{R}^{+*} \times \mathbb{R}^{+*}$, *if* $(q,t,\omega_1) \in Tag_{\mathcal{Q},\Delta}$ *and* $(q,t,\omega_2) \in Tag_{\mathcal{Q},\Delta}$ *then* $\omega_1 = \omega_2$.
- *Full covering tagging: each question is tagged with at least one tag and is therefore useful.*
- *Frugal tagging: a question is tagged with at most one tag and is therefore considered only once.*
- *Leaf tagging: questions are tagged by, and only by tags which are leafs of the dimension of analysis.*
- *No orphan tag: each leaf tag of the analysis dimension is used in at least one tagging.*

Notice that the property of frugal tagging may be satisfied by defining two different questions with the same formulation.

Starting from a given analysis dimension, a well-formed information need can be obtained simply by associating questions to each of its leaves and only to its leaves. Since a question is only associated to one node, this amounts to completing the analysis dimension tree by adding a level of questions to the leaves.

For the rest of the formalization, we will only consider well-formed information needs as it guarantees that all elements, questions and tags, are useful and well connected with each others.

Example 3.6 (Simple well-formed information need example)**.**
Let us consider $\mathcal{Q} = \{q_0, \ldots, q_9\}$ *a synthetic set of questions. Figure 5 schematizes a possible information need* $\mathcal{N}_{6W} = \langle \mathcal{Q}, \Delta_{6W}, Tag_{\mathcal{Q},\Delta_{6W}} \rangle$ *with 6Ws dimension.*

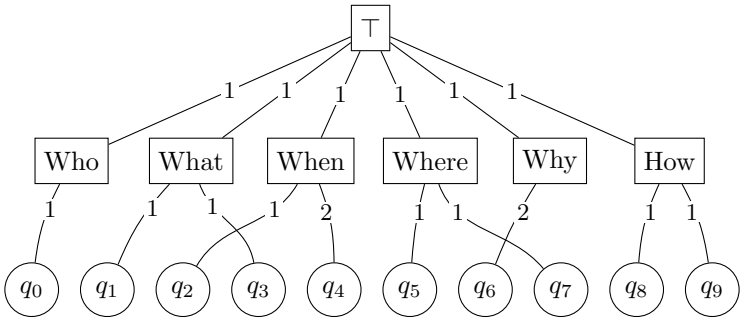


Figure 5: Information Need w.r.t. 6Ws Analysis Dimension

Example 3.7 (A well-formed information need to analyze a photograph)**.**
Let $\mathcal{Q}_P$ *be a set containing the following questions:*

q_1: *Who is the author of the photograph?*

q_2: *What is the title of the photograph?*

q_3: *When was taken the photograph?*

q_4: *What is the exposure of the photograph?*

q_5: *What are the main colors?*

q_6: *What are the leading lines?*

q_7: *What type is it (portrait, landscape...)?*

q_8: *What is the main subject?*

q_9: *Are there any people?*

q_{10}: *What can be seen on the photograph?*

q_{11}: *What is the historical context?*

q_{12}: *What were the influences of the photograph?*

q_{13}: *What is the message of the photographer?*

q_{14}: *Why was it taken?*

These questions can be tagged using the analysis dimension Δ_P, introduced in example 3.2, in order to organize them. Then the need $\mathcal{N}_P = \langle \Delta_P, \mathcal{Q}_P, Tag_{Q_P,\Delta_P} \rangle$ is defined as shown in the Figure 6.

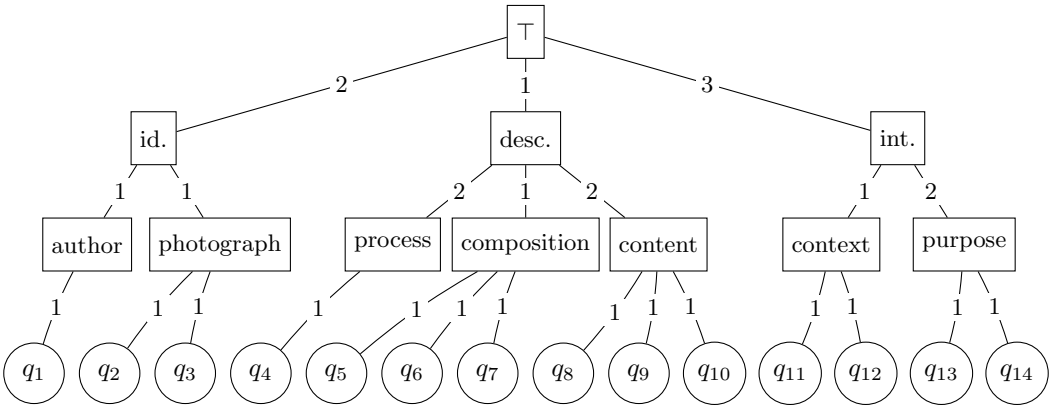


Figure 6: Need for photography analysis

3.2.2 Operators on information needs

As for the analysis dimensions it is possible to introduce operators to manipulate information needs. For the sake of simplicity, the number of operators presented here is limited to two. None of the presented operator proposes direct manipulation on the question set nor on the tagging. The eventual modifications of these elements will therefore be exclusively the consequences of the modifications of the analysis dimensions.

The first operator presented is the restriction of a well-formed need. It implements the focus on a precise part of the need (forgetting the other parts) which echoes the restriction of a dimension (see definition 3.2). So, the restriction of a well-formed need is naturally based on the restriction of the analysis dimension that composes it. When a tag is removed, the questions associated with it are also forgotten. A difficulty arises from the fact that a user could ask to retain a node while not retaining any of its children, which amounts to asking to forget all the elements that define a tag in the need without forgetting the tag itself. Although such a specification may be considered inconsistent, we have chosen to allow the user to express it. In this case, it is up to the operator to overcome this problem (which makes it a little more difficult to express) and produces a well-formed need.

Definition 3.7 (Restricting a well-formed need)**.**
Let $\mathcal{N} = \langle \mathcal{Q}, \Delta, Tag_{\mathcal{Q},\Delta} \rangle$ be a well-formed need. Let S be a non empty set such that $S \subseteq T_\Delta$. A ***restriction*** *of $\mathcal{N}$ w.r.t. S, noted $\mathcal{N}|_S = \langle \mathcal{Q}', \Delta', Tag_{\mathcal{Q}',\Delta'} \rangle$ is defined by:*

- $\Delta' = \Delta|_{S_{wf}}$ *with*

$$S_{wf} = \{t \mid t \in T_{\Delta|_S} \text{ and } ((t \text{ is a leaf in } \Delta) \text{ or } (t \text{ has a descendant in } \Delta|_S \text{ which is a leaf in } \Delta)\}$$

- $\mathcal{Q}' = \{q \mid q \in \mathcal{Q} \text{ and } \exists (t, \omega) \in T_{\Delta'} \times \mathbb{R}^{+*}, (q, t, \omega) \in Tag_{\mathcal{Q},\Delta}\}$
- $Tag_{\mathcal{Q}',\Delta'} = \{(q, t, \omega) \mid (q, t, \omega) \in Tag_{\mathcal{Q},\Delta} \text{ and } t \in T_{\Delta'}\}$

Note that S_{wf} is obtained restricting S_S is such a way that each leaf in Δ' was already a leaf in Δ, i.e. the restriction does not transform any node of the initial analysis dimension in a leaf.

Theorem 3.3.
Let $\mathcal{N} = \langle \mathcal{Q}, \Delta, Tag_{\mathcal{Q},\Delta} \rangle$ be a well-formed need. Let S be a non empty subset of T_Δ, then $\mathcal{N}|_S$ is a well-formed need.

Proof. To prove the result to be a well-formed need, we have first to prove it's a need.

- $\mathcal{Q}'$ is a set defined as a subset of $\mathcal{Q}$ which by hypothesis is a set of questions. So $\mathcal{Q}'$ is also a set of questions.
- Δ' is obtained form a dimensions using an operator over dimension. Trivial.
- $Tag_{\mathcal{Q}',\Delta'}$ by construction $Tag_{\mathcal{Q}',\Delta'} \subset \mathcal{Q}' \times T_{\Delta'} \times \mathbb{R}^{+*}$.

So $\mathcal{N}|_S$ is a need.
Now, let us prove $\mathcal{N}|_S$ is well-formed.

Unicity of weights Let $q \in \mathcal{Q}'$, $t \in T_{\Delta'}$ and $\omega_1, \omega_2 \in \mathbb{R}^{+*}$ such that $(q, t, \omega_1) \in Tag_{\mathcal{Q}',\Delta'}$ and $(q, t, \omega_2) \in Tag_{\mathcal{Q}',\Delta'}$. Since $Tag_{\mathcal{Q}',\Delta'} \subseteq Tag_{\mathcal{Q},\Delta}$ by construction, and $\mathcal{N}$ is well-formed by assumption, it follows that $\omega_1 = \omega_2$. Consequently, $\mathcal{N}|_S$ satisfies the property of unicity of tags.

Full covering tagging Let $q \in \mathcal{Q}'$. By construction of $\mathcal{Q}'$, $\exists (t, \omega) \in T_{\Delta'} \times \mathbb{R}^{+*}$ such that $(q, t, \omega) \in Tag_{\mathcal{Q},\Delta}$. As $t \in T_{\Delta'}$, we deduce that $(q, t, \omega) \in Tag_{\mathcal{Q}',\Delta'}$. Therefore, q is tagged with at least one tag of $T_{\Delta'}$ and $\mathcal{N}|_S$ satisfies the property of full covering.

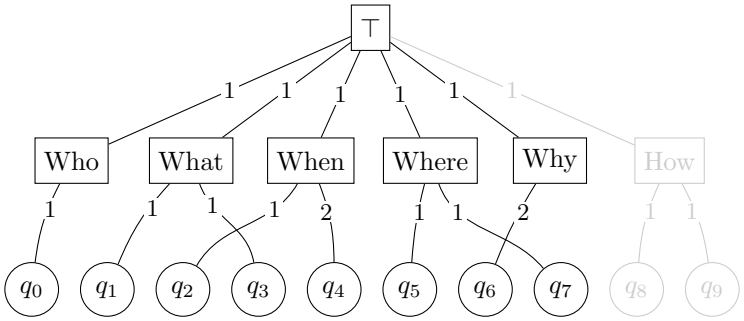
Frugal tagging Let us assume the resulting tagging being not a frugal tagging. So, there exists one question $q \in \mathcal{Q}'$ which is tagged with two tags t and t'. Since by construction $Tag_{\mathcal{Q}',\Delta'}$ is a restriction of $Tag_{\mathcal{Q},\Delta}$, these two are also part of the original need, which contradicts the hypothesis that it is well-formed.

Leaf tagging Let us assume the resulting tagging being not a leaf tagging. So, there exists a triple $(q, t, \omega) \in Tag_{\mathcal{Q}',\Delta'}$ such that t is not a leaf in Δ'. By construction, $(q, t, \omega) \in Tag_{\mathcal{Q},\Delta}$ and since by hypothesis, $\mathcal{N}$ is a well-formed need, t is a leaf in Δ'. Since Δ' is obtained from Δ by restriction, either t is no longer in Δ' or it is still a leaf, which contradicts t is not a leaf in Δ'.

No orphan tag Let us assume that the resulting tagging has at least one orphan tag. Then, $\exists t \in T_{\Delta'}$ such that t is a leaf in Δ' and there does not exist (q, ω) such that $(q, t, \omega) \in Tag_{\mathcal{Q}',\Delta'}$. As far as $\mathcal{N}$ is well-formed, this means that t is not a leaf in Δ. The two elements "t is not a leaf in Δ" and "t a leaf in Δ'" contradict the fact that t is in S_{wf} by definition of S_{wf}.

□

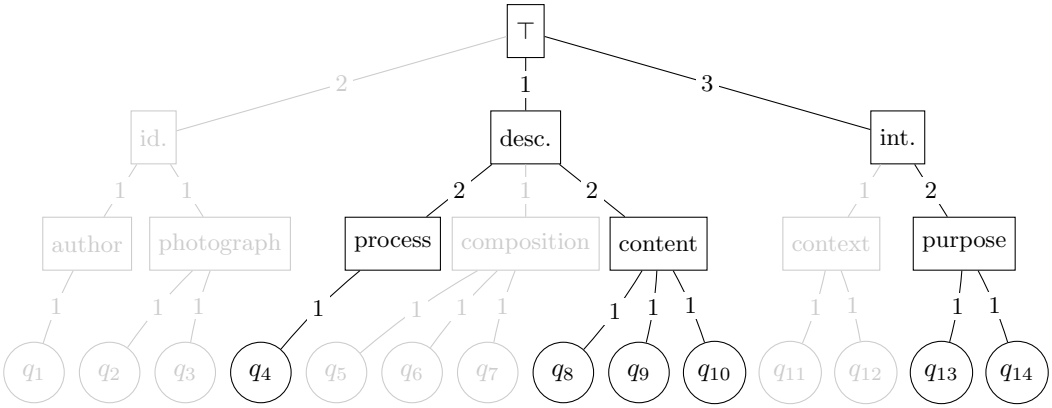
Example 3.8 (Restricting the 6Ws well-formed need)**.**
Let us consider the previous 6Ws dimension associated with its need $\mathcal{N}_{6W}$. A restriction of this need w.r.t. the 5Ws, is the need $\mathcal{N}_{5W}$ illustrated by the Figure 7. It results that some questions, q_8 and q_9, are not tagged anymore and are then removed, they appear in grey in the figure.



In light grey, the elements that are no longer part of the resulting need.

Figure 7: Restricting 6Ws Need w.r.t. 5Ws Analysis Dimension: $\mathcal{N}_{6W}|_{T_{\Delta_{6W}}\setminus\{How\}}$

Example 3.9 (Restricting the information need to analyze a photograph)**.**
Let us consider the well-formed information need $\mathcal{N}_P$ about the photograph analysis, and the same restriction as in the example 3.4. A restriction of this need $\mathcal{N}_P|_{\{id.,desc.,process,content,purpose\}}$ is presented in Figure 8. Compared to $\Delta_P|_{\{id.,desc.,process,content,purpose\}}$, the tag "id." is removed because it is not a leaf in Δ_P, and has no descendant in the restricted dimension.



In light grey, the elements that are no longer part of the resulting need.

Figure 8: Restriction of the Photograph Analysis Information need

In some cases, it may be interesting to assess the loss due to some restriction w.r.t. the initial need. This can be done quite simply, thanks to the weights. Here, we focus on the question loss.

Definition 3.8 (Coverage rate of a restriction)**.**
*Let $\mathcal{N} = \langle \mathcal{Q}, \Delta, Tag_{\mathcal{Q},\Delta} \rangle$ be a well-formed information need and $\mathcal{N}|_S = \langle \mathcal{Q}', \Delta', Tag_{\mathcal{Q}',\Delta'} \rangle$ a restriction of it. The **coverage rate** of $\mathcal{N}|_S$ compared to $\mathcal{N}$ determines the weighted coverage of the questions kept in $\mathcal{N}|_S$ compared to the whole questions of $\mathcal{N}$. It is noted $cr(\mathcal{N}|_S, \mathcal{N})$ and is defined as:*

$$cr(\mathcal{N}|_S, \mathcal{N}) = cr(\mathcal{N}|_S, \mathcal{N}, \top)$$

With

$$cr(\mathcal{N}|_S, \mathcal{N}, t) = \begin{cases} \frac{\sum_{t'|tR_{\Delta'}t'} \delta_\Delta(t') \times cr(\mathcal{N}|_S, \mathcal{N}, t')}{\sum_{t'|tR_\Delta t'} \delta_\Delta(t')} & \text{if } t \text{ is not a leaf in } \Delta' \\ 1 & \text{if } t \text{ is a leaf in } \Delta \text{ and } \Delta' \\ 0 & \text{otherwise} \end{cases}$$

The ***loose rate*** *of $\mathcal{N}|_S$ compared to $\mathcal{N}$ is defined as:*

$$lr(\mathcal{N}|_S, \mathcal{N}) = 1 - cr(\mathcal{N}|_S, \mathcal{N})$$

In fact, the coverage rate focuses on the leaf tags of the analysis dimension of the need. Indeed, for each tag, either all questions are preserved, or all are removed. Hence, this definition can also be used for leaf coverage rate of the analysis dimension.

Example 3.10 (Coverage rate of the six Ws restriction)**.**
The coverage rate of $\mathcal{N}_{5W}$ compared to $\mathcal{N}_{6W}$ is simply $\frac{5}{6}$.

Example 3.11 (Coverage rate of the restriction of the information need to analyze a photograph)**.**
Let us consider the restriction of the information need on photograph analysis. Let X be the set of preserved tags, $X = \{id., desc., process, content, purpose\}$. Then the coverage rate of $\mathcal{N}_P|_X$ compared to $\mathcal{N}_P$ is defined by: $cr(\mathcal{N}_P|_X, \mathcal{N}_P) = cr(\mathcal{N}_P|_X, \mathcal{N}_P, \top)$. Where:

$$\begin{aligned} cr(\mathcal{N}_P|_X, \mathcal{N}_P, \top) &= \frac{\sum_{t'|tR_{\Delta'}t'} \delta_\Delta(t') \times cr(\mathcal{N}_P|_X, \mathcal{N}_P, t')}{\sum_{t'|tR_\Delta t'} \delta_\Delta(t')} \\ &= \frac{1 \times cr(\mathcal{N}_P|_X, \mathcal{N}_P, desc.) + 3 \times cr(\mathcal{N}_P|_X, \mathcal{N}_P, int.)}{2 + 1 + 3} \\ &= \frac{1 \times 4/5 + 3 \times 2/3}{6} \\ &= \frac{7}{15} \end{aligned}$$

Where id. is not considered in the numerator of the formula because it is not a child of $\top$ in Δ' ; $cr(\mathcal{N}_P|_X, \mathcal{N}_P, desc.)$ and $cr(\mathcal{N}_P|_X, \mathcal{N}_P, int.)$ are obtained by applying the formula once again, where the leaves preserved are given the value 1.

As for analysis dimension, we propose an operator to expand a well-formed need $\mathcal{N}_1 = \langle \mathcal{Q}, \Delta_1, Tag_{\mathcal{Q},\Delta_1} \rangle$ with another $\mathcal{N}_2 = \langle \mathcal{Q}, \Delta_2, Tag_{\mathcal{Q},\Delta_2} \rangle$ to obtain a third one $\langle \mathcal{Q}, \Delta, Tag_{\mathcal{Q},\Delta} \rangle$. Naturally, the set of questions $\mathcal{Q}$ has to be the same between the two needs and is not modified. Δ is defined by the operator on analysis dimensions ($\Delta = \Delta_1 \lhd \Delta_2$). In order to combine two needs, their dimensions should be independent.

To be well-formed, the need must only use leaf tags of $\Delta_1 \lhd \Delta_2$. Hence, a question is only tagged with a leaf of $\Delta_1 \lhd \Delta_2$ if it is tagged with the corresponding node of Δ_2 and of Δ_1. In order to consider the relative importance of the question relatively with its two tags, the resulting weight it the multiplication of the two initial weights.

A tricky point is that, if left unattended, the resulting analysis dimension could have leafs without associated questions. The operator has to overcome this difficulty. To do so, it removes those parts of the analysis dimension that are orphaned in terms of questions.

Definition 3.9 (Extending a well-formed need with another)**.**
*Let $\mathcal{N}_1 = \langle \mathcal{Q}, \Delta_1, Tag_{\mathcal{Q},\Delta_1} \rangle$ and $\mathcal{N}_2 = \langle \mathcal{Q}, \Delta_2, Tag_{\mathcal{Q},\Delta_2} \rangle$ be two well-formed needs associated with of the same set of questions $\mathcal{Q}$, Δ_1 and Δ_2 being independent analysis dimensions. The **extension** of $\mathcal{N}_1$ with $\mathcal{N}_2$, noted $\mathcal{N}_1 \lhd \mathcal{N}_2 = \langle \mathcal{Q}, \Delta, Tag_{\mathcal{Q},\Delta} \rangle$ is defined by:*

- $\Delta = (\Delta_1 \lhd \Delta_2)|_{S_{wf}}$ *with*

  $$S_{wf} = \{t_1.t_2 \mid \exists(q, \omega_1, \omega_2) \text{ such that } ((q, t_1, \omega_1) \in Tag_{\mathcal{Q},\Delta_1} \text{ and } (q, t_2, \omega_2) \in Tag_{\mathcal{Q},\Delta_2})\}$$

  Reminder: $t_1.t_2$ is the pointed notation to identify different instances of Δ_2 in $\Delta_1 \lhd \Delta_2$ (see definition 3.3).

- $Tag_{Q,\Delta} = \{(q,\ t_1.t_2,\ \omega_1 \times \omega_2) \mid (q, t_1, \omega_1) \in Tag_{Q,\Delta_1} \text{ and } (q, t_2, \omega_2) \in Tag_{Q,\Delta_2}\}$

Theorem 3.4.
Let $\mathcal{N}_1 = \langle \mathcal{Q}, \Delta_1, Tag_{\mathcal{Q},\Delta_1} \rangle$ and $\mathcal{N}_2 = \langle \mathcal{Q}, \Delta_2, Tag_{\mathcal{Q},\Delta_2} \rangle$ be two well-formed needs associated with of the same set of questions $\mathcal{Q}$, Δ_1 and Δ_2 being independent analysis dimensions. $\mathcal{N}_1 \lhd \mathcal{N}_2$ is a well-formed need.

Proof. First, let us prove that $\mathcal{N}_1 \lhd \mathcal{N}_2$ is a need.

- $\mathcal{Q}$ is a set of questions. Trivial.
- $\Delta = (\Delta_1 \lhd \Delta_2)|_{S_{wf}}$ is an analysis dimension. Trivial as used operators take dimensions and produce dimensions, as far as S_{wf} is not empty.

  By definition and hypothesis, $\mathcal{Q}$ is non empty. Let us consider a question q of $\mathcal{Q}$. Since $\mathcal{N}_1$ is well-formed, q is tagged with some leaf of Δ_1, i.e. $\exists(q, t_1, \omega_1) : (q, t_1, \omega_1) \in Tag_{\mathcal{Q},\Delta_1}$. Similarly, $\exists(q, t_2, \omega_2) : (q, t_2, \omega_2) \in Tag_{\mathcal{Q},\Delta_2}$. Then, by construction, $(t_1.t_2) \in S_{wf}$, which means that $S_{wf} \neq \emptyset$.

- $Tag_{\mathcal{Q}',\Delta} \subset \mathcal{Q} \times T_\Delta \times \mathbb{R}^{+*}$ Let us consider a triple (q, t, ω) belonging to $Tag_{\mathcal{Q},\Delta}$.
  - $q \in \mathcal{Q}$. Indeed, by construction, q is obtained from triples of $Tag_{\mathcal{Q},\Delta 1}$ and $Tag_{\mathcal{Q},\Delta_2}$ which, by hypothesis tags questions.
  - $t \in T_\Delta$. By definition, if t satisfies the conditions for (q, t, ω) to be part of T_Δ, t also satisfies the condition to be part of S_{wf}, so $t \in T_{\Delta_1 \lhd \Delta_2)|_{S_{wf}}}$, i.e. $t \in T_\Delta$.
  - $\omega \in \mathbb{R}^{+*}$. Trivial : $\omega = \omega_1 \times \omega_2$ and $\omega_1 \in \mathbb{R}^{+*}$ and $\omega_2 \in \mathbb{R}^{+*}$.

Now, let us prove that $\mathcal{N}_1 \lhd \mathcal{N}_2$ is a well-formed need.

Unicity of weights Let $q \in \mathcal{Q}$, $t \in T_\Delta$ and $\omega, \omega' \in \mathbb{R}^{+*}$ such that $(q, t, \omega) \in Tag_{\mathcal{Q},\Delta}$ and $(q, t, \omega') \in Tag_{\mathcal{Q},\Delta}$. Then, there exists t_1 in T_{Δ_1}, t_2 in T_{Δ_2}, and $\omega_1, \omega_2, \omega_1', \omega_2' \in \mathbb{R}^{+*}$ such that $(q, t_1, \omega_1), (q, t_1, \omega_1') \in Tag_{\mathcal{Q},\Delta_1}$ and $(q, t_2, \omega_2), (q, t_2, \omega_2') \in Tag_{\mathcal{Q},\Delta_2}$ with $t = t_1.t_2$, $\omega = \omega_1 \times \omega_2$ and $\omega' = \omega_1' \times \omega_2'$. Since $\mathcal{N}_1$ and $\mathcal{N}_2$ are well-formed, then $\omega_1 = \omega_1'$ and $\omega_2 = \omega_2'$. Therefore $\omega_1 = \omega_2$ and $\mathcal{N}_1 \lhd \mathcal{N}_2$ satisfies the property of unicity of tags.

Full covering tagging Let $q \in \mathcal{Q}$. As $\mathcal{N}_1$ and $\mathcal{N}_2$ are well-formed, then there is t_1 in T_{Δ_1}, t_2 in T_{Δ_2} and $\omega_1, \omega_2 \in \mathbb{R}^{+*}$ such that $(q, t_1, \omega_1) \in Tag_{\mathcal{Q},\Delta_1}$ and $(q, t_2, \omega_2) \in Tag_{\mathcal{Q},\Delta_2}$. By construction, $(q,\ t_1.t_2,\ \omega_1 \times \omega_2) \in Tag_{Q,\Delta}$. Therefore, q is tagged in with at least one tag of T_Δ and $\mathcal{N}_1 \lhd \mathcal{N}_2$ satisfies the property of full covering.

Frugal tagging Let $q \in \mathcal{Q}$ such that q is tagged with two tags t, t' of $\in T_\Delta$. Then, $\exists t_1, t'_1 \in T_{\Delta_1}$ and $\exists t_2, t'_2 \in T_{\Delta_2}$ such that $t = t_1.t_2$, $t' = t'_1.t'_2$, q is tagged by t_1 and t'_1 in $\mathcal{N}_1$ and q is tagged by t_2 and t'_2 in $\mathcal{N}_2$. Since $\mathcal{N}_1$ and $\mathcal{N}_2$ are well-formed, then $t_1 = t'_1$ and $t_2 = t'_2$ and therefore $t = t'$ and q is tagged with only one tag in $Tag_{Q,\Delta}$. And $\mathcal{N}_1 \lhd \mathcal{N}_2$ satisfies the property of frugal tagging.

Leaf tagging Let $q \in \mathcal{Q}$. Then, q is tagged by $t_1 \in T_{\Delta_1}$ in Δ_1 and by $t_2 \in T_{\Delta_2}$ in Δ_2, where t_2 is a leaf tag because $\mathcal{N}_2$ is well-formed. As $\mathcal{N}_2$ is well-formed, t_2 is a leaf tag of Δ_2. Then, by construction, $t_1.t_2$ is a leaf in $\Delta_1 \lhd \Delta_2$, and it is preserved in $\Delta = (\Delta_1 \lhd \Delta_2)|_{S_{wf}}$. Therefore q is tagged with a leaf in Δ, and $\mathcal{N}_1 \lhd \mathcal{N}_2$ satisfies the property of leaf tagging.

No orphan tag Let t be a leaf of Δ. $t \in S_{wf}$ because if t is not in S_{wf} then it would have been removed from Δ. So, $\exists (t_1, t_2) \in T_{\Delta_1} \times T_{\Delta_2}$ such that $t = t_1.t_2$ there exists $q \in \mathcal{Q}$ tagged by t_1 in Δ_1 and by t_2 in Δ_2. Then q is tagged by t. Therefore, each leaf of Δ is used in at least one tagging and $\mathcal{N}_1 \lhd \mathcal{N}_2$ has no orphan tag.

□

Example 3.12 (Extending needs)**.**
Let us consider only the first level of the Photograph Analysis Dimension Δ_{P1} extended with the 5Ws Analysis Dimension $\Delta_{P1} \lhd \Delta_{5W}$. Hence, Figure 9 shows how to combine their needs into $\mathcal{N}_{P1} \lhd \mathcal{N}_{5W}$. Tags appearing in grey in the figure are the ones removed from the need compared to the original analysis dimension $\Delta_{P1} \lhd \Delta_{5W}$. The weights of an element in the Figure 9c are computing by multiplying the weights of the element in the Figure 9a and in the Figure 9b. Notice that q_5 and q_7 share the same relative importance w.r.t. the tag Where, and q_5 is three times more important than q_7 w.r.t. "int.". Hence, q_5 is still three times more important than q_7 w.r.t. the couple of tags ("int.", Where).

3.3 Completeness

In order to measure to what extend the knowledge graph answers the questions forming the need, we introduce completeness. It should take into account the structure of the need, i.e. its analysis dimension. There are several ways to do so. The most natural one, using all information at our disposal, is to use a weighted average.

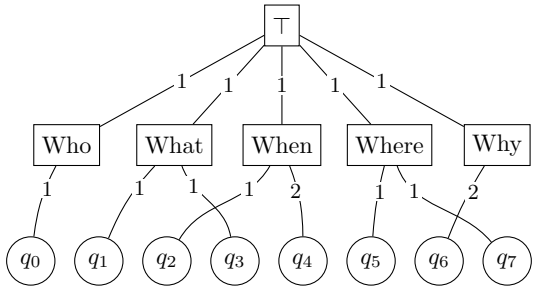
Definition 3.10 (Evaluation of KG completeness w.r.t. an information need)**.**
Let $\mathcal{N} = \langle \mathcal{Q}, \Delta, Tag_{\mathcal{Q},\Delta} \rangle$ be a well-formed information need, and g a knowledge graph. A quantified evaluation of the ***completeness*** *of g w.r.t. $\mathcal{N}$ is obtained as follows:*

$$completeness(g, \mathcal{N}) = completeness(g, \mathcal{N}, \top)$$

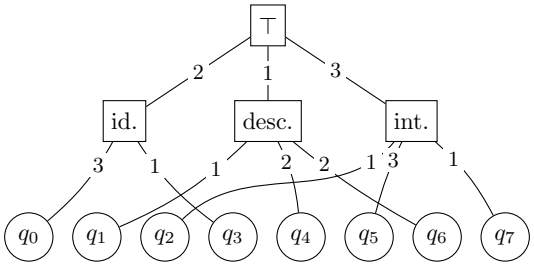
for a given tag t

$$completeness(g, \mathcal{N}, t) = \begin{cases} \dfrac{\sum_{t' | t R_\Delta t'} \delta_\Delta(t').completeness(g, \mathcal{N}, t')}{\sum_{t' | t R_\Delta t'} \delta_\Delta(t')} & \text{if } t \text{ is not a leaf.} \\ \dfrac{\sum_{q,\omega | (q,t,\omega) \in Tag_{\mathcal{Q},\Delta}} \omega.evalAnswer(q,g)}{\sum_{q,\omega | (q,t,\omega) \in Tag_{\mathcal{Q},\Delta}} \omega} & \text{if } t \text{ is a leaf.} \end{cases}$$

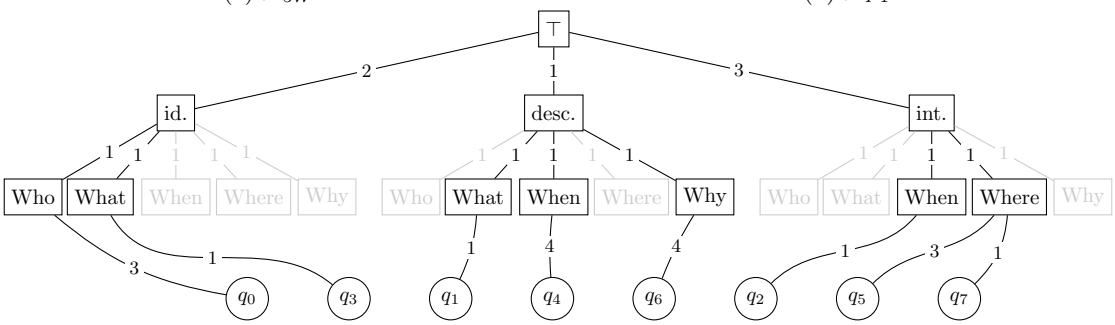
where $evalAnswer(q, g)$ is assumed to provide a normalized evaluation of the answer provided by g regarding the question q.



(a) $\mathcal{N}_{5W}$



(b) $\mathcal{N}_{P1}$



In light grey, the elements that are no longer part of the resulting need.

(c) Visualizing extension of $\mathcal{N}_{P1}$ with $\mathcal{N}_{5W}$: $\mathcal{N}_{P1} \lhd \mathcal{N}_{5W}$

Figure 9: Extending one need with another

Example 3.13 (Evaluation of completeness w.r.t. the need for photography analysis)**.** *Let $\mathcal{N}_P = \langle \mathcal{Q}_P, \Delta_P, Tag_{\mathcal{Q}_P,\Delta_P} \rangle$ be the need of interest, and g be a knowledge graph about a given photograph. The first step to compute the completeness of g regarding $\mathcal{N}_P$ is to compute its value of completeness on all leaf tags.*

For instance, on "purpose":

$$
\begin{aligned}
completeness(g, \mathcal{N}_P, purpose) &= \frac{\sum\limits_{q,\omega|(q,purpose,\omega)\in Tag_{\mathcal{Q}_P,\Delta_P}} \omega.evalAnswer(q,g)}{\sum\limits_{q,\omega|(q,purpose,\omega)\in Tag_{\mathcal{Q}_P,\Delta_P}} \omega} \\
&= \frac{\omega_{q_{13}}.evalAnswer(q_{13},g) + \omega_{q_{14}}.evalAnswer(q_{14},g)}{\omega_{q_{13}} + \omega_{q_{14}}} \\
&= \frac{evalAnswer(q_{13},g) + evalAnswer(q_{14},g)}{2}
\end{aligned}
$$

If the questions are answered, then the value is 1*, if only one of them is answered, the value is* 0.5 *and* 0 *otherwise.*

Then, the second step consists in computing the completeness of g on all the parents tags of

leaf tags. Hence, for "int.", we obtain:

$$completeness(g, \mathcal{N}_P, int.) = \frac{\sum_{t' | int.R_{\Delta_P} t'} \delta_{\Delta_P}(t').completeness(g, \mathcal{N}_P, t')}{\sum_{t' | int.R_{\Delta_P} t'} \delta_{\Delta_P}(t')}$$
$$= \frac{completeness(g, \mathcal{N}_P, context) + 2.completeness(g, \mathcal{N}_P, purpose)}{3}$$

Then, the process continues on all their parents tags, until the root is treated.

3.4 Implementation

The previous definition of completeness is general. This section aims to complete it in order to provide at an evaluation that can be directly exploited and automated with semantic web tools. To do this, we first focus on the implementation of an information need. In a second step, we are interested in its representation by proposing an ontology that includes the concepts that make it up.

3.4.1 Implementing information needs

Two new elements are introduced by the implemented need: first, the translation of the questions into queries in order to make the questions executable on the knowledge graph and second the explicit definition of the *evalAnswer* function which evaluates the results.

Definition 3.11 (Implemented need)**.**
*An **implemented need**, noted* $\mathtt{N}$*, is a tuple* $\mathtt{N} = \langle \Delta, \mathcal{Q}, Tag_{\mathcal{Q},\Delta}, i, evalAnswer \rangle$ *such that*

- $\langle \Delta, \mathcal{Q}, Tag_{\mathcal{Q},\Delta} \rangle$ *is a well-formed information Need.*
- *i is an implementation function such that $i(q)$ provides an executable implementation of the question q, or "undefined" if the question is not implementable. The implementation can be a SPARQL query for instance.*
- *evalAnswer computes a normalized evaluation, of the result of the execution of an implementation $i(q)$ of a question q on a KG g. The evaluation is 0 is the implementation is "undefined". With $q \in \mathcal{Q}$, and g a KG, $evalAnswer(q, g) \in [0, 1]$.*

*A **well-implemented need** is an implemented need such that every question is implementable, i.e. there is no q such that $i(q) =$ "undefined".*

Unfortunately, in some situations it is not possible to implement a question. For instance, a question can be too imprecise or too subjective. It is also possible that the question uses concepts not present in the ontologies or tools used to build the KG. In all these cases the implementation function just returns "*undefined*".

3.4.2 An ontology for information need

With the objective to take advantage of the Semantic Web technologies, we represent the concept of information need and the elements constituting it in RDF. We define the ontology SIN-O[2] as illustrated by Figure 10. As a reminder, an information is defined as follows $\mathcal{N} = \langle \mathcal{Q}, \Delta, Tag_{\mathcal{Q},\Delta} \rangle$. Hence, an *InformationNeed* is composed of a set of questions, pictured on the right, of an analysis dimension which is a set of structured tags, pictured on the left, and of links between these two sets represented by labelings, i.e. a tagging, pictured in the middle.

[2]Available at: `https://github.com/Jendersen/KG_accountability/tree/main/information_need`

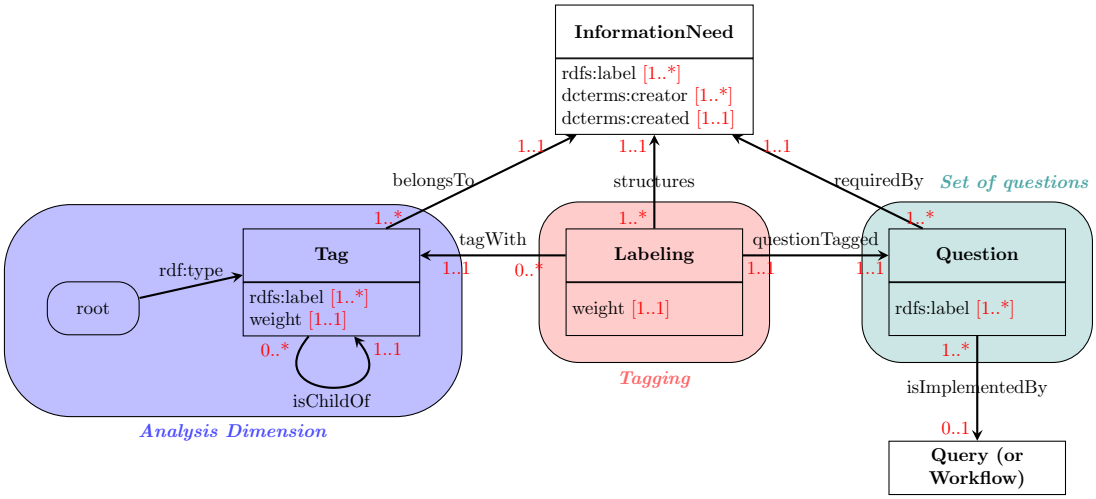


Figure 10: SIN-O: ontology of the information need

In addition, SHACL constraints have been added to the ontology in order to verify if the need is well-formed. Currently, these constraints are expressed considering only one need. In the Figure 10, they are represented with the cardinalities in red (for unicity of weights, full covering tagging and frugal tagging). Some other constraints check if a *Tag* either has a child or is used in a *Labeling* (leaf tagging and no orphan tag) and if each tag is a descendant of the *root* (the analysis dimension is a tree rooted at *root*).

Finally, to avoid defining inconsistent or unexpected analysis dimensions, we only allow a single *InformationNeed* to be associated with a *Tag*. Indeed there is only one structuring possible with this representation because the property *isChildOf* is only related with the tag and does not depend on the information need. As *Labeling* is associated with a tag, it can only belong to one analysis dimension. As a result, tags and questions cannot be used in several information needs, they must be copied in order to be reused.

4 Assessment of Knowledge Graphs Accountability

As underlined in section 2.1, Transparency and Accountability are very much linked. Indeed, the latter may be seen as being transparent about information linked to people responsibility and justification, or verification, of data use and misuse. In this section, we illustrate how to define an information need and compute the completeness of several KGs in order to provide an assessment of their accountability. In this way, our approach offers a first insight into the transparency of KGs.

Our starting point is the LiQuID metadata model [16]. It provides a comprehensive list of questions that need to be answered to judge the accountability of a dataset. The question set is also structured through a three-level tree. LiQuID main elements are presented in the following subsection. The analysis dimension we propose for assessing the accountability of knowledge graphs is adapted from the structure proposed in LiQuID. The questions are also adapted and specified. Then, questions are translated into queries in order to define the implemented need. Finally, we evaluate KGs using these queries and we deduce their completeness regarding this need.

4.1 The LiQuID metadata model of Accountability

4.1.1 General presentation

The information need we propose is based on the LiQuID metadata model, a well-known work from Oppold and Herschel [16]. It provides a representation of information related to accountability in order to enable datasets to improve on that aspect. This work is not specific to any type of dataset so it can be adapted to KGs.

The LiQuID metadata model relies on a hierarchical structure. First, it covers all steps of a dataset's life cycle: data collection, processing, maintenance, and usage. Each life cycle step is structured according to different question types: why, who, when, where, how, and what (cf. the 6W dimension in the example 3.1). Finally, each question type is divided into different fields of information level: description, explanation, legal and ethical considerations, and limitations. The authors provide an exhaustive list of questions to describe each aspect of this hierarchy. For instance, for "Data Collection", question type "Why?", the question associated with the field "Description" is "Why was the data set created?" and for the field "Ethical Considerations": "Why is it ethical to create a data set for this cause?". There can also be more than one question per field. LiQuID proceeds in a very systematic way and requires a large amount of very detailed information, representing what data sources should expose to be as accountable as possible.

4.1.2 LiQuID as a base to define an Information need

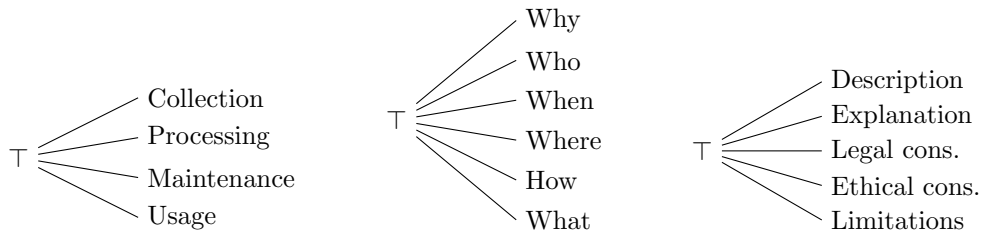
There are several ways to define information needs depending on the way the LiQuID hierarchical structure and associated questions is considered.

Need based on the whole LiQuID structure. It is quite natural to define an information need based on LiQuID. Its hierarchy provides an analysis dimension Δ_{LiQuID} represented in Figure 11. Only weights are missing: LiQuID does not mention any weight nor any relative importance of some elements over others. Then, we arbitrarily decided that all tags are of equal importance, which we represent by assigning the value 1 to the weight of any tag. Questions given by LiQuID define the set of questions $\mathcal{Q}_{LiQuID}$. In LiQuID, each question is associated with one and only one leaf tag of the analysis dimension. This directly defines the basis of the tagging Tag_{LiQuID}. However, again, no weights are mentioned. Here also, we arbitrarily we assign the value 1 to the weight of each question to indicate that we consider them all equally important. Hence, $\mathcal{N}_{LiQuID} = \langle \mathcal{Q}_{LiQuID}, \Delta_{LiQuID}, Tag_{LiQuID} \rangle$ has naturally the properties of a well-formed need: unicity of weights, full covering tagging and leaf tagging are satisfied. Furthermore, there is at least one question per field, so there is no orphan tag. And even though some questions are present in multiple tags, they implicitly depend on the context so they can be considered different questions. Therefore the property of frugal tagging is also verified.

More elementary needs. Moving away from the original presentation of LiQuID, each level of its hierarchy can be represented as a specific need. Indeed, every question is simultaneously associated with one step of the lifecycle, one question type, and one field of the information level. Let $\mathcal{N}_{LifeCycle}$, $\mathcal{N}_{QuestionType}$, $\mathcal{N}_{InfoLevel}$ be respectively the need associated with the life cycle level, the question level and the information level. Their analysis dimensions are presented in Figure 12. These needs are well-formed for the same reasons as mentioned before. The original LiQuID need can easily be obtained by combining them. Indeed, it is the extension of $\mathcal{N}_{LifeCycle}$ with $\mathcal{N}_{QuestionType}$ and then extended with $\mathcal{N}_{InfoLevel}$: $\mathcal{N}_{LiQuID} = \mathcal{N}_{LifeCycle} \lhd \mathcal{N}_{QuestionType} \lhd \mathcal{N}_{InfoLevel}$. This way of building LiQuID is very instinctive as the analysis dimensions are independent.

One may want to define additional needs for further analysis. It is then very convenient to use $\mathcal{N}_{QuestionType}$, $\mathcal{N}_{LifeCycle}$, $\mathcal{N}_{InfoLevel}$ and the restriction and extension operations. For instance,

Figure 11: LiQuID Analysis Dimension Δ_{LiQuID}



(a) Life cycle level $\Delta_{LifeCycle}$ (b) Question level $\Delta_{QuestionType}$ (c) Information level $\Delta_{InfoLevel}$

Figure 12: Three analysis dimensions to obtain LiQuID

one could change the order of extensions: $\mathcal{N}_{QuestionType} \lhd \mathcal{N}_{LifeCycle} \lhd \mathcal{N}_{InfoLevel}$; or focus on a specific need only, such as $\mathcal{N}_{QuestionType}$; or even restrict the need to some tags only, e.g. all questions of type "Who", no matter the step of the life cycle. In fact, these three needs act as a generator of needs, saving the work of designing new structures from scratch. In addition it is possible to analyze the results in many angles, with no additional graph querying. In such cases, it is sufficient to compute $evalAnswer(q, g)$ once on each question of the needs to obtain $completeness(g, \mathcal{N}_{LiQuID})$ or every other variation.

4.2 Definition of a specific need $\mathcal{N}_{AK}$ for Knowledge Graphs

In order to remain as close as possible to the initial approach of LiQuID, we have chosen to base our definition of a specific need $\mathcal{N}_{AK}$ on $\mathcal{N}_{LiQuID}$.

4.2.1 Adaptation of $\mathcal{N}_{LiQuID}$

Ideally, to assess the accountability of a KG, we should consider $\mathcal{N}_{LiQuID}$ and all the questions should be implemented into SPARQL queries. However, this is not possible. Firstly, as shown by Oppold and Herschel, the two general metadata models Dublin Core[3] and PROV [13], used in KGs, "even combined they only cover 51.7%" of the fields proposed by LiQuID. Indeed, both models "contain few fields, some of them too general to be mapped to specific LiQuID fields" [16]. We make the same observation with other general metadata models used in KGs. As a consequence of this lack of expressivity, sometimes, either a question cannot be translated into a query, or two questions result in the same query for different steps of the life cycle. Secondly, some questions had to be made more precise and adapted to the context of Knowledge Graphs.

For these reasons, combined with some arbitrary simplification choices and the objective to obtain a well-formed and well-implemented need $\mathcal{N}_{AK}$, our adaptation of LiQuID impacts both the analysis dimension, by removing some tags, and the questions.

The analysis dimension Δ_{AK} of $\mathcal{N}_{AK}$ *(i)* only considers the tag "description" of the information level and, *(ii)* does not consider the data processing step of the life cycle level nor the tags 'why", "data collection.what" and "data maintenance.what". In other words, dimension Δ_{AK} is obtained by restriction of the analysis dimension of LiQuID: $\Delta_{AK} = \Delta_{LiQuID}|_X$, where X is the set of all tags from LiQuID except those that have just been mentioned. The result of this restriction is shown in Figure 13.

The questions we consider are derived from the ones of LiQuID. In addition, some questions are divided into smaller parts, so they focus on only one element each. This precision is made as faithfully as possible, with the aforementioned limitations. All LiQuID questions of the field "description" of the three life cycle steps considered are listed in Tables 1, 2 and 3 with their associated questions.

In total, LiQuID contains 207 questions. When only considering the analysis dimension Δ_{AK}, it remains 25 LiQuID questions. Notice that most of the questions that we do not consider come from the data processing step and, above all, from the other fields of the information level (157). These fields require very detailed information and do not seem reasonable to consider regarding the content of the existing KGs. This loss can also be seen with the coverage rate, the value of which is $cr(\mathcal{N}_{LiQuID}, \mathcal{N}_{AK}) = \frac{13}{120}$. Implementing the remaining LiQuID questions would be an interesting future work to provide a more complete need on accountability.

The 25 LiQuID questions left are used to define more precise questions and result into 32 questions: 6 for Data Collection, 6 for Data Maintenance and 20 for Data Usage. The questions are listed in the Tables 1, 2 and 3. The need of information is then defined as follows. Let

[3] `https://dublincore.org/specifications/dublin-core/dcmi-terms/`

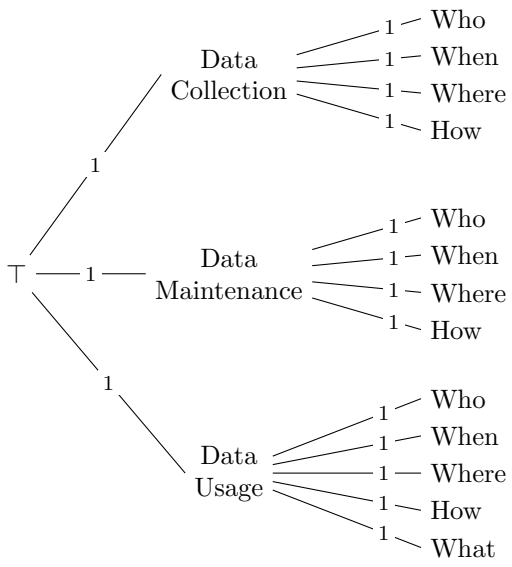


Figure 13: The analysis dimension Δ_{AK} of a first need on accountability

$\mathcal{N}_{AK} = \langle \mathcal{Q}_{AK}, \Delta_{AK}, Tag_{\mathcal{Q}_{AK},\Delta_{AK}} \rangle$ be the need, with $\mathcal{Q}_{AK}$ the set of 32 questions obtained and Δ_{AK} defined before. Then, the tagging $Tag_{\mathcal{Q}_{AK},\Delta_{AK}}$ is inherited from LiQuID. When a LiQuID question, tagged by t, is made more precise by another unique question, then the latter is also tagged by t and with the same weight of 1. When a LiQuID question is expressed by n more precise questions, these are all tagged by t and their weights take the value $1/n$.

4.2.2 Implementation of $\mathcal{N}_{AK}$

Once the information need has been defined, questions can be translated into SPARQL queries or successions of SPARQL queries. In order to write these queries, ten vocabularies of reference are chosen regarding their pertinence to describe datasets and concepts around: VoID [1] is used to express metadata about RDF datasets. DCAT[4] and DataID[5] allow the description of datasets and catalog of datasets. SPARQL-SD [25] enables to describe SPARQL endpoints. These vocabularies rely on other general vocabularies, the Dublin Core[6] and FOAF[7]. We also use PROV-O [13] and PAV [7] for provenance issues. DQV[8] is used to describe the quality of datasets. Finally, we use schema.org[9] a very general and widely used vocabulary. Each query uses all coherent properties and classes of these vocabularies to be as complete as possible. Listing 1 shows an example of question translated into a query, where `?kg` must be replaced by the IRI of the knowledge graph at hand.

Listing 1: Query associated with "Who publishes this dataset?"

```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX dce: <http://purl.org/dc/elements/1.1/>
```

[4] https://www.w3.org/ns/dcat
[5] http://dataid.dbpedia.org/ns/core
[6] https://dublincore.org/specifications/dublin-core/dcmi-terms/
[7] http://xmlns.com/foaf/spec/
[8] https://www.w3.org/TR/vocab-dqv/
[9] https://www.schema.org/

Table 1: Questions associated with Data Usage

Tag	Questions from LiQuID	Questions adapted to KG	ω
Usage. Why	What has the data set been used for?	(1)	1
	For which other purposes can the published data set be used for?	(1)	1
Usage. Who	Who publishes this data set?	Who publishes this KG?	1
	Who has used/ can use the published data set?	Who has the right to use the published KG?	1/2
		Who is intended to use the published KG?	1/2
Usage. When	When can/ was the published data set be used?	Since when was the KG available?	1
	When is it available?	Until when is the KG available?	1
	Until what point in time is it valid?	Until when is the KG valid?	1
Usage. Where	Where is the data set published/ available?	What is the webpage presenting the KG and/or allowing to gain access to it?	1/2
		Where to access the KG (either through a dump or a SPARQL endpoint)?	1/2
	Where (place, geographically) can the published data set be used?	In what physical location can the KG be used?	1
Usage. How	What is a recommended process for using the published data set?	What is the license of the KG?	1/3
		How to access the KG? Provide a SPARQL endpoint or a dump if they are freely accessible, or the procedure of access, and the characteristics of the endpoint if provided.	1/3
		How to use, reuse or integrate the KG?	1/3
	What are recommended methods, tools, and technical environments where the published data set can be used?	What are the requirements to use the KG?	1
Usage. What	What data is published for use?	What are examples of the published data?	1
	What concepts does it cover?	What concepts, topics or subjects does the KG cover?	1
	What is a general description of the data set?	What is a general description of the KG?	1
	What are the characteristics/ profile of the data set (dependent on data type)?	How many triples are there in the KG?	1/3
		How many entities, properties and classes are there in the KG?	1/3
		What RDF serialization formats does the KG support?	1/3
	What is the quality of the data set (quality metrics depend on data type)?	What is the quality of the KG?	1

(1) Vocabularies miss expressivity

```
PREFIX schema: <http://schema.org/>
PREFIX prov: <http://www.w3.org/ns/prov#>
ASK {
    {?kg dcterms:publisher ?publisher .}
    UNION {?kg dce:publisher ?publisher .}
    UNION {?kg schema:publisher ?publisher .}
    UNION {?kg schema:sdPublisher ?publisher .}
```

```
    UNION {?kg prov:wasGeneratedBy ?act .
           ?act a prov:Publish .
           ?act prov:wasAssociatedWith ?publisher .}
}
```

The 32 questions were translated into 30 main queries or succession of queries. Indeed, some queries induce new queries, for instance when the result of a question triggers other questions. For example, the "Data Collection/Who" question which first looks for creators and then asks multiple things about each of them. It is then necessary to get the result of the first query in order to execute the following ones. Furthermore, two questions were not translated into a query because of the complexity of their translation. As an example, let us consider the question "Which methods and tools were exactly used in each step and what was the (technical) environment?". It is very difficult to transform it into query (or queries) due to the multiple valid ways to represent provenance in KGs. For a first evaluation campaign, it does not appear reasonable to us to treat this question. It could be the subject of a future work focused on provenance. So in total, 39 queries are written to express the need.

The implemented need $\mathtt{N}_{AK} = \langle \Delta_{AK}, \mathcal{Q}_{AK}, Tag_{\mathcal{Q}_{AK},\Delta_{AK}}, i, evalAnswer \rangle$ is derived from $\mathcal{N}_{AK}$, where i associates a SPARQL query or a workflow of queries to each question and "undefined" to the two aforementioned questions. Moreover, *evalAnswer* returns 1 in case (i) the execution of an "ASK" query results in TRUE or (ii) the execution of a "SELECT" query has at least one result. Otherwise *evalAnswer* returns 0. For a workflow, the queries are organized as a tree, where each query depends on the result of the previous query (its parent in the tree). Therefore, from bottom to top, *evalAnswer* computes for each node the average of the result of the current node with the results obtained before for its children.

4.3 Evaluation of the Completeness of KGs w.r.t. $\mathcal{N}_{AK}$

As $\mathcal{N}_{AK}$ is implemented, we can assess the completeness of a KG, as defined in section 3.3. Using it enables to propose a first automatic measure of accountability of a KG. In a first step, we describe the method employed to conduct an evaluation campaign of several KGs. Then we analyze the results. All our queries and results are publicly available on a Github repository[10].

4.3.1 Method

To evaluate several knowledge graphs, we use the framework IndeGx[11]. It indexes public KGs accessible through SPARQL endpoints with a SPARQL-based test suit. The process uses SPARQL queries to extract or compute metadata about the KG and produces a description of the KG. We embed the set of queries obtained in section 4.2.2 into the framework. As they are associated with questions requiring answers, they are ASK queries. The answer TRUE is considered a success as it means that the KG contains the desired information while an answer FALSE or an error (e.g., timeout exception) is a failure as it means the KG is not able to provide the wanted information. The result of the test is then added to the description of the studied KG.

Our experiments evaluate the KGs already identified by IndeGx, meaning the ones with endpoints listed on the LOD Cloud, Wikidata, SPARQLES, Yummy Data and Linked Wiki. In total, 670 knowledge graphs are evaluated.

A prerequisite of all our queries is to identify the IRI that the studied KG uses for itself. Indeed, all our queries are looking for metadata about the KG, such as the author, the license, etc. To do so, it is necessary to know this IRI, which is the subject of at least one triple in all our

[10] https://github.com/Jendersen/KG_accountability
[11] https://github.com/Wimmics/dekalog

queries. This is done with a preliminary query presented in Listing 2, where `$rawEndpointUrl` is provided by IndeGx and represents the URL of the endpoint under evaluation. If the KG does not provide an answer to this query, it will not answer any of our queries. Hence, this query is used to select the KGs that will be evaluated on the whole need.

Among the 670 KGs studied, IndeGx sometimes study both the URL starting by `http` and the one starting by `https`. While the associated KG is the same, the result of Listing 2 may differ, and therefore the results on the whole set of queries also.

Listing 2: Preliminary Query to identify the IRI of the studied KG

```
PREFIX void: <http://rdfs.org/ns/void#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>
SELECT ?kg_iri WHERE {
        { ?kg_iri a dcat:Dataset }
        UNION { ?kg_iri a void:Dataset }
        ?kg_iri ?endpointLink $rawEndpointUrl .
}
```

This preliminary query of Listing 2 is executed on all KGs at three different dates and hours, separated by several days. The objective is to detect all candidates and not to penalize them if they are not available during the period of evaluation. All KGs succeeding this query at least once are selected for the next step. It consists in evaluating each query of the need $\mathtt{N}_{AK}$, executing it three times, still at different time points. For each KG, only the results of the last experiment for which it was available are kept. Doing so, we do not penalize a KG which was unavailable at some time point and we favor the last version which should the most up to date. We did not encounter the problem of a KG becoming unavailable during the querying phase.

Given the results obtained for each query by IndeGx, we successively compute *evalAnswer* of each question of the need $\mathcal{N}_{AK}$. We then aggregate the results to compute the completeness related to each tag of the information need and then the completeness of KGs w.r.t. the whole need.

4.3.2 Results

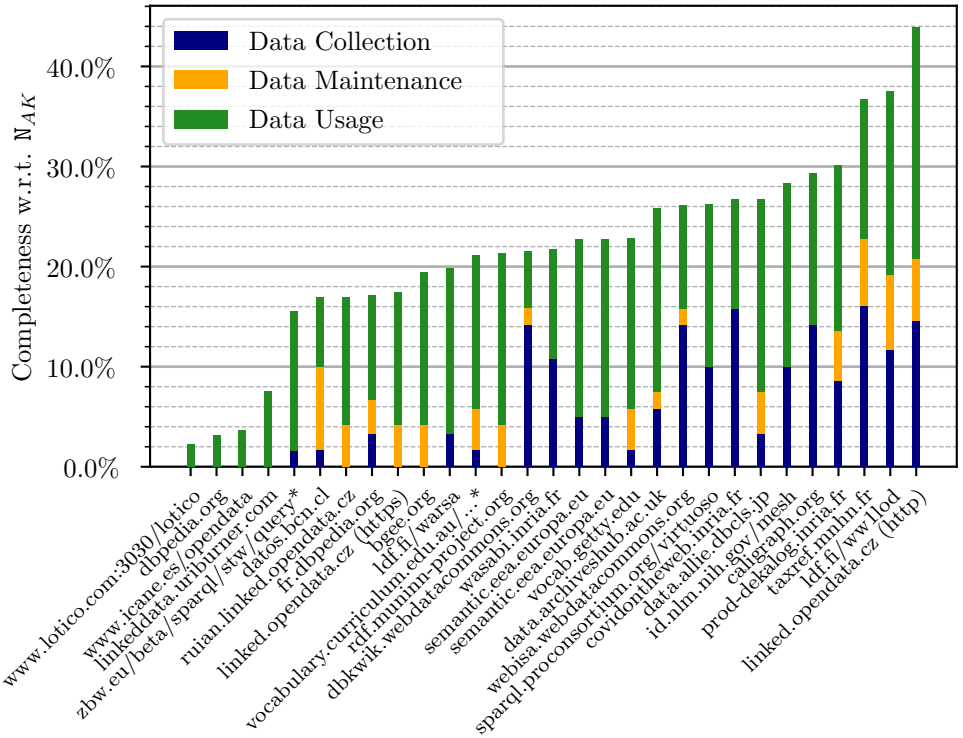
Only a few KGs provides metadata about themselves. Among the 670 KGs tested, only 29 successfully pass the preliminary query (Listing 2). While this is quite few, this result was to be expected. Indeed, according to KartoGraphI[12] [14], based on IndeGx, only a few KGs provide a description of themselves, 2 to 6%, depending on the type of description. Hence, most of the KGs do not provide metadata about themselves within their own data, therefore it is useless for them to provide their own IRI and most of them do not provide one.

Regarding our evaluation, one must keep in mind that some meta-information may be provided by KG producers outside of the KG itself, for instance on its webpage. It can also be inside the KG but not related to the URL of the endpoint nor to an entity of type Dataset from VoID or DCAT. In both cases, the meta-information is not detected and therefore not considered. While this may penalize KGs that provide information related to accountability, it points out the fact that they are less transparent because information is less accessible.

However, this cannot hold for all the KGs that failed the preliminary query. It shows that there is still a lot of work to do for knowledge graphs to be even at least a little accountable. Even if the required information is very common, it seems that most of the KGs do not provide metadata within their data. For providers who fear to mix these two kind of data, a good practice would be to separate data from metadata using named graphs, with one dedicated to metadata.

[12] `http://prod-dekalog.inria.fr/`

Completeness of the 29 successful KGs Even though most of the KGs have a completeness value of zero, the need on accountability allows to discriminate between the 29 KGs left, with values distributed between 2.2% and 44%. The mean and median of these values are of 22%. Considering the life cycle steps, completeness w.r.t. "data collection" is between 0% and 48%, w.r.t. "data maintenance" is between 0% and 25% and w.r.t. "data usage" is between 6.7% and 69%. On average, KGs are twice more complete on Data Usage than on Data Collection, and about 2.5 times better on Data Collection than on Data Maintenance.

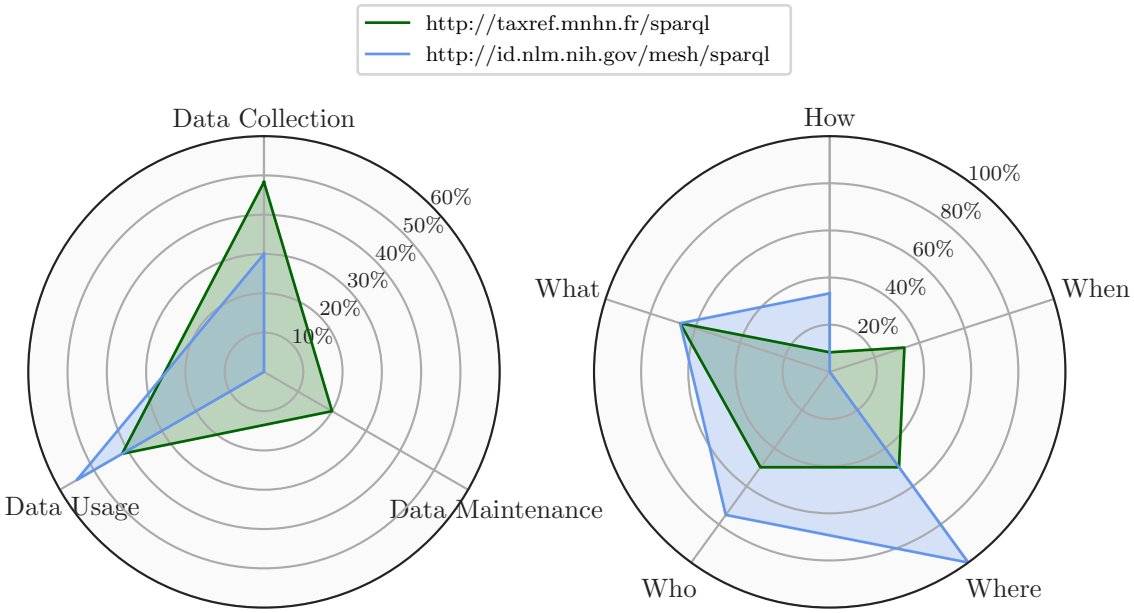


All URL start with `http://` or `https://` and, except the ones with *, end with `/sparql`.

Figure 14: Completeness of KGs w.r.t. the need N_{AK}

In details, Figure 14 shows the measures of completeness regarding N_{AK} of the KGs. They are divided into the three main tags "data collection", "data maintenance" and "data usage". As the weights of each tag are equal, the completeness regarding each tag is their height on the scale, divided by three (the number of tags). For instance, the completeness of `taxref.mnhn.fr` and `id.nlm.nih.gov/mesh` on these dimensions is shown on Figure 15a. It is also possible to observe the completeness w.r.t. more precise tags. Hence, Figure 15b compares the two KGs on the tags of Data Usage and shows that `id.nlm.nih.gov/mesh` is better on "data usage.who", "data usage.where", "data usage.how".

When analyzing the results on the different lifecycle steps, most of the KGs have a greater value on "data usage" than on the other steps. This may be explained by two factors. First, there are more questions in "data usage" than elsewhere, therefore there is a better chance to have one of the required information. That can also explain the fact that no KG has a value of 0. Then, the information asked in "data usage" is more general: a general description, a link to



(a) Completeness of two KGs w.r.t. the lifecycle tags

(b) Completeness of two KGs w.r.t. the tags of Data Usage

Figure 15: Completeness of two KGs w.r.t. different tags of $\mathtt{N}_{AK}$

a sparql endpoint, a license... and therefore more commonly provided.

Given the fact that most of the KGs have a value of 0, we consider that a completeness above the mean, 22%, is a good score of accountability. The KG with the highest completeness is `http://linked.opendata.cz/sparql`. However, the evaluated endpoint is linked with almost a hundred different datasets, and for each question, if at least one of them provides a required information, the query is considered a success. This may explain the good result. Notice that most of the evaluated endpoints are linked with only one dataset. The best KG linked with only one dataset is `http://taxref.mnhn.fr/sparql`. In particular, it relies on the best score on "data collection" and one of the best on "data maintenance".

Even though the completeness values are quite low, all KGs have a margin to improve themselves considering what the different KGs are able to answer. Indeed, a KG succeeding all queries which were answered at least once would have a completeness of 61%. For "data collection", it is 62.5%, "data maintenance": 50% and for "data usage" it is 71%. Therefore, considering that some KGs are able to answer this, it is easy to improve on collection and maintenance.

Analysis of the information need $\mathtt{N}_{AK}$ Concerning the need itself, it is important to notice that some queries never succeed, 9 out of 30. Especially, the tags "data collection.how", "data maintenance.how", and "data maintenance.where" always get a value of 0. The distribution of the values of completeness w.r.t. each tag is represented in Figure 16. Each box represents the first quartile (Q1), the median and the third quartile (Q3) of the values obtained on the tag and the whiskers indicate the minimum and the maximum values obtained. Sometimes, the median and the first quartile are equal as in "data usage.where". It shows that the tags of "data usage" have usually good values in the different KGs. It also shows that only three tags can be fully covered: "data collection.when", "data maintenance.when" and "data usage.where".

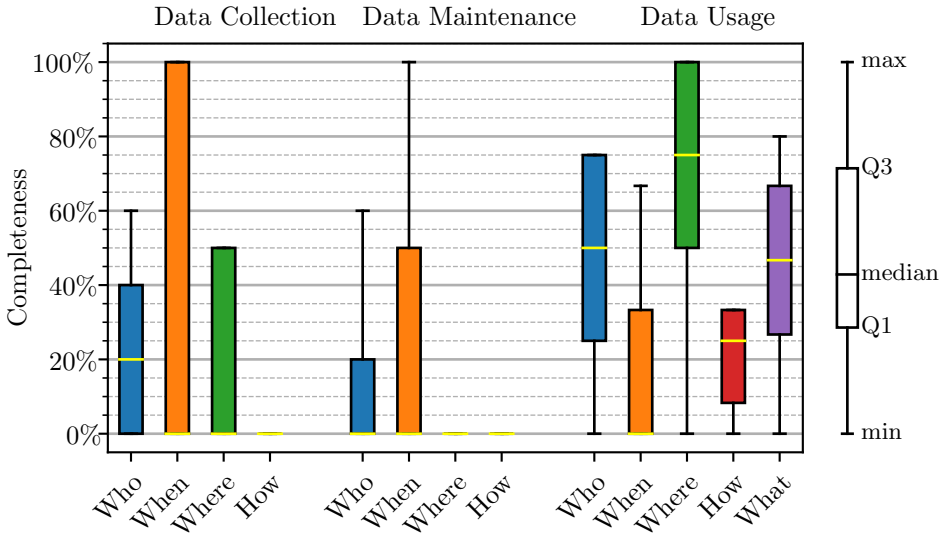


Figure 16: Box plot representing the distribution of the values of completeness w.r.t. each tag

Several reasons may explain that some queries are never answered. Either KGs do not provide enough information. Or answers are too difficult to obtain because the query is too complicated or expressed with too few vocabularies to hope for an answer. If neither of them can explain the lack of answers, then it may be interesting to analyze how relevant is the question associated with the query.

5 Discussions

To summarize, we defined an information need, to which was associated a measure of completeness. Therefore, it is measured regarding an explicit need which enhance the understandability of the measure as well as its transparency. Both to illustrate how to use the information need and to provide a first measure of accountability, we defined an information need focused on accountability. Then we ran an experiment on several Knowledge Graphs to assess their completeness w.r.t. accountability, which can also be seen as a first and partial score of transparency.

Our approach was initially designed for the use case of transparency. More generally, it intends to cover all needs of information within the knowledge graph, whether they be prototypical and widely shared or very specific. Notice that the classical approach of data quality evaluation can also be expressed using the information need as we defined it. For instance, FAIR is naturally structured and provide a list of requirements that can be transformed into a list of questions. However, FAIR does not totally fit the spirit of our need as it mainly focuses on technical aspects and do not always look for information within the dataset.

For future work, it would be interesting to provide a tool to help data consumer designing their own need and generated the associated RDF graph in order to easily share it and reuse it. Furthermore, new operators could be defined to transform any need into a well-formed need.

We measured the completeness of several KGs w.r.t. the need focused on accountability. While too many KGs get a score of zero because they did not provide any metadata about themselves,

the measure still discriminate among 27 KGs. There is room for a lot of improvement, and guidelines may play an important role to make KGs provide metadata within their data (in a specific named graph for instance), and to make them provide minimal information (contributors, creation and modification dates, license...) as asked in metadata completeness [12].

A few queries of our need are never answered. Indeed, some of these queries are probably too difficult or with too few vocabularies to hope for an answer. For instance, while provenance is of major importance to understand how data was obtained, it is complicated to query this due to the multiple ways of representing this. A future work to increase the number of queries answered would be to consider more vocabularies. However, there could still be some queries without answers. For these, it would be interesting to analyze, with experts and KG providers, to what extent the required information is effectively relevant in the context of semantic web. Depending on the result, it is possible to modulate the question to make it more general or more suitable to KGs. Or, if this is not enough, the question can be assigned a lower weight or it can be removed.

Our measure of completeness relies on the hypothesis that "incorrect data values do not adversely affect the assessment" [12]. Furthermore, we are satisfied if there is at least one answer to each query. However, there probably is more than one contributor and we would benefit from the information that all contributors are provided. This additional information, discussed in [8], is complementary with our work and would improve the completeness assessment.

Contents

References

[1] Alexander, K., Cyganiak, R., Hausenblas, M., and Zhao, J. Describing linked datasets with the VoID vocabulary. *W3C Note. W3C* (2011).

[2] Amdouni, E., and Jonquet, C. Une méthodologie et un outil d'évaluation du niveau de "FAIRness" pour les ressources sémantiques: le cas d'agroportal. In *Jourrnées francophones d'Ingénierie des Connaissances (IC 2021)* (2021).

[3] Amdouni, E., and Jonquet, C. Fair or fairer? an integrated quantitative fairness assessment grid for semantic resources and ontologies. In *Research Conference on Metadata and Semantics Research* (2022), Springer, pp. 67–80.

[4] Andersen, J., Cazalens, S., and Lamarre, P. On the way to measure KG transparency: formalizing transparency - requirements and first models - DeKaloG D31. Tech. rep., 2021.

[5] Basili, V. R., Caldiera, G., and Rombach, H. D. The goal question metric approach. *Encyclopedia of software engineering* (1994), 528–532.

[6] Bertino, E. The quest for data transparency. *IEEE Security & Privacy 18*, 3 (2020), 67–68.

[7] Ciccarese, P., Soiland-Reyes, S., Belhajjame, K., Gray, A. J., Goble, C., and Clark, T. Pav ontology: provenance, authoring and versioning. *Journal of biomedical semantics 4*, 1 (2013), 1–22.

[8] Darari, F., Razniewski, S., Prasojo, R. E., and Nutt, W. Enabling fine-grained rdf data completeness assessment. In *International Conference on Web Engineering* (2016), Springer, pp. 170–187.

[9] Debattista, J., Lange, C., Auer, S., and Cortis, D. Evaluating the quality of the lod cloud: An empirical investigation. *Semantic Web 9*, 6 (2018), 859–901.

[10] Färber, M., Bartscherer, F., Menne, C., and Rettinger, A. Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web 9*, 1 (2018), 77–129.

[11] Firmani, D., Tanca, L., and Torlone, R. Ethical dimensions for data quality. *Journal of Data and Information Quality (JDIQ) 12*, 1 (2019), 1–5.

[12] Issa, S., Adekunle, O., Hamdi, F., Cherfi, S. S.-S., Dumontier, M., and Zaveri, A. Knowledge graph completeness: A systematic literature review. *IEEE Access 9* (2021), 31322–31339.

[13] Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Corsar, D., Cheney, J., Garijo, D., Soiland-Reyes, S., Zednik, S., and Zhao, J. PROV-O: The PROV Ontology. *W3C Recommendation. W3C* (2013).

[14] Maillot, P., Corby, O., Faron, C., Gandon, F., and Michel, F. KartoGraphI: Drawing a Map of Linked Data. In *ESWC 2022 - 19th European Semantic Web Conferences* (Hersonissos, Greece, May 2022), Springer.

[15] Matheus, R., and Janssen, M. Transparency dimensions of big and open linked data. In *Conference on e-Business, e-Services and e-Society* (2015), Springer, pp. 236–246.

[16] Oppold, S., and Herschel, M. Accountable data analytics start with accountable data: The liquid metadata model. In *ER Forum/Posters/Demos* (2020), pp. 59–72.

[17] Rosnet, T., de Lamotte, F., Devignes, M.-D., Lefort, V., and Gaignard, A. FAIR-checker - supporting the findability and reusability of digital life science resources.

[18] Schultes, E., Magagna, B., Hettne, K. M., Pergl, R., Suchánek, M., and Kuhn, T. Reusable fair implementation profiles as accelerators of fair convergence. In *International Conference on Conceptual Modeling* (2020), Springer, pp. 138–147.

[19] Schwabe, D., Laufer, C., and Casanovas, P. Knowledge graphs: Trust, privacy, and transparency from a legal governance approach. *Law in Context. A Socio-legal Journal 37*, 1 (2020), 1–19.

[20] Vandenbussche, P.-Y., Umbrich, J., Matteis, L., Hogan, A., and Buil-Aranda, C. Sparqles: Monitoring public sparql endpoints. *Semantic web 8*, 6 (2017), 1049–1065.

[21] Wang, R. Y., and Strong, D. M. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems 12*, 4 (1996), 5–33.

[22] Weitzner, D. J., Abelson, H., Berners-Lee, T., Feigenbaum, J., Hendler, J., and Sussman, G. J. Information accountability. *Communications of the ACM 51*, 6 (2008), 82–87.

[23] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. The FAIR guiding principles for scientific data management and stewardship. *Scientific data 3*, 1 (2016), 1–9.

[24] Wilkinson, M. D., Sansone, S.-A., Schultes, E., Doorn, P., Bonino da Silva Santos, L. O., and Dumontier, M. A design framework and exemplar metrics for FAIRness. *Scientific data 5*, 1 (2018), 1–4.

[25] Williams, G. T. Sparql 1.1 service description. *W3C Recommendation. W3C* (2013).

[26] Wyatt, D. The many dimensions of transparency: A literature review. *Helsinki Legal Studies Research Paper*, 53 (2018).

[27] Yamamoto, Y., Yamaguchi, A., and Splendiani, A. Yummydata: providing high-quality open life science data. *Database 2018* (2018).

[28] Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., and Auer, S. Quality assessment for linked data: A survey. *Semantic Web 7*, 1 (2016), 63–93.

Table 2: Questions associated with Data Collection

Tag	**Questions from LiQuID**	**Questions adapted to KG**	ω
Collection. Why	Why was the data set created?	*	1
Collection. Who	Who (people, organizations) was involved in the data collection process? Provide all information relevant to their identification, their role in the data collection process, all information necessary to assess their qualifications to fulfill this role, and all characteristics which could have an influence on the data set.	Who are the creators of the KG and their role in this process? For all creators, indicates whether they are a person or an organization, provide information to identify them (name and point of contact such as email, or phone number, or address, or homepage), provide their qualifications, provide all characteristics which could have an influence on the KG.	1
Collection. When	On what date(s) or time frame(s) has the data been collected/ created? It must also be possible to place the data in a temporal context.	What is/are the creation date(s) of the KG?	1
Collection. Where	Where was the data set collected (country, place, website, ...)? It must also be possible to place the data in a spatial context.	From what original source(s) were the data collected or derived?	1/2
		From what physical location (state, country, continent, ...) was the KG created?	1/2
Collection. How	What was the methodology/ procedure for data collection?	Which methods or tools were used for data creation?	1
	Which methods and tools were exactly used in each step and what was the (technical) environment?	Which methods and tools were exactly used in each step and what was the (technical) environment? **	1
Collection. What	What data was collected?	*	1
	What concepts does it cover?	*	1
	What is a general description of the data set?	*	1
	What are the characteristics/ profile of the data set (dependent on data type)?	*	1
	What is the quality of the data set (quality metrics depend on data type)?	*	1

* Vocabularies miss expressivity to distinguish between collected data and published data (cf. Table 1)
** This question has no query associated

Table 3: Questions associated with Data Maintenance

Tag	**Questions from LiQuID**	**Questions adapted to KG**	ω
Maintenance. Why	Why will the dataset be further maintained?	*	1
Maintenance. Who	Who (people, organizations) will be involved in the data maintenance? Provide all information relevant to their identification, their role in the data maintenance, all information necessary to assess their qualifications to fulfill this role, and all characteristics which could have an influence on the data set.	Who are the maintainers of the KG and their role in this process? For all maintainers, indicates whether they are a person or an organization, provide information to identify them (name and point of contact such as email, or phone number, or address, or homepage), provide their qualifications, provide all characteristics which could have an influence on the KG.	1
Maintenance. When	On what date(s) or time frame(s) will the data be maintained?	When was the KG last maintained/-modified?	1
	With which frequency?	With which frequency is the KG maintained?	1
Maintenance. Where	Where will the data set be maintained (country, place, website, . . .)?	From what physical location (state, country, continent, ...) is or will the KG be maintained?	1
Maintenance. How	What will be the methodology/ procedure for data maintenance?	What will be the methodology/ procedure for data maintenance?	1
	Which methods and tools will exactly be used in each step and what will be the (technical) environment?	Which methods and tools will exactly be used in each step and what will be the (technical) environment? **	1
Maintenance. What	What data will be the result of the data maintenance?	*	1
	What concepts does it cover?	*	1
	What is a general description of the data set?	*	1
	What are the characteristics/ profile of the data set (dependent on data type)?	*	1
	What is the quality of the data set (quality metrics depend on data type)?	*	1

* Vocabularies miss expressivity to distinguish between maintained data and published data (cf. Table 1)
** This question has no query associated