Is my:sameAs the same as your:sameAs?
Lenticular Lenses for Context-Specific Identity

Al Koudous Idrissou¹,², Rinke Hoekstra¹, Frank van Harmelen¹, Ali Khalili¹
¹ Department of Computer Science,
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
o.a.k.idrissou@vu.nl

Peter van den Besselaar²
² Department of Organization Sciences,
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands

ABSTRACT
Linking between entities in different datasets is a crucial element of the Semantic Web architecture, since those links allow us to integrate datasets without having to agree on a uniform vocabulary. However, it is widely acknowledged that the owl:sameAs construct is too blunt a tool for this purpose. It entails full equality between two resources independent of context. But whether or not two resources should be considered equal depends not only on their intrinsic properties, but also on the purpose or task for which the resources are used. We present a system for constructing context-specific equality links. In a first step, our system generates a set of probable links between two given datasets. These potential links are decorated with rich metadata describing how, why, when and by whom they were generated. In a second step, a user then selects the links which are suited for the current task and context, constructing a context-specific “Lenticular Lens”. Such lenses can be combined using operators such as union, intersection, difference and composition. We illustrate and validate our approach with a realistic application that supports researchers in social science.

CCS CONCEPTS
• Computing methodologies → Knowledge representation and reasoning;

KEYWORDS
owl:sameAs, linkset, lens, data integration

1 INTRODUCTION
Constructing and maintaining links between corresponding entities in different datasets and ontologies is a crucial element of the Semantic Web architecture. After all, these links are responsible for the integration of datasets and ontologies published by multiple independent parties without the requirement to agree a priori on a uniform vocabulary, allowing the Semantic Web to scale.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

K-CAP 2017, Knowledge Capture Conference, December 4–6, 2017, Austin, TX, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5555-7/17/12...$15.00
https://doi.org/10.1145/3148011.3148029

The quality of such correspondence links depends on how well the requirements for the correspondence fit the formal semantics of the predicate used to express it. Linking across datasets for information retrieval purposes poses different constraints than use cases where statistical analysis or automated decision making play a role. And similarly, a false positive skos:related relation has less far reaching consequences than a misguided owl:sameAs. As long as the data is used in isolation, and the links are well understood, this problem is manageable. However, when data and links are shared on the Web, the context in which a link was deemed to be true is lost. This is especially problematic for transitive predicates that formally state equality between resources. For instance, [9] shows that owl:sameAs is often misused. As you follow a chain of owl:sameAs triples, the stated equivalence between the two outer ends of the chain becomes increasingly tenuous.¹ At the same time [3] argues that the use of weaker alternatives such as rdfs:seeAlso reduces the utility of the relation to express the intended semantics. It would seem that different linktypes are required in different circumstances. Or rather, that the truth value of an equality assertion depends on the context for which it was generated.

We can illustrate the problems with the semantics of owl:sameAs with an example from a scientific domain that performs analysis across multiple datasets: the field of Science, Technology and Innovation (STI) studies the dynamics of scientific ideas [8]. For this, the field depends on a large variety of heterogeneous data.

To study the successful collaboration of scientific organizations, STI researchers need to align such organizations across datasets that describe organisations across various countries, e.g. GRID² and OrgRef³. The 3M corporation, a large multinational organisation with a substantial patent portfolio, occurs in both datasets. GRID distinguishes between national 3M branches across six countries, while OrgRef only refers to a single 3M entity. Should these entities be designated as “the same” across these datasets? It depends. For a study that aims to compare organizations at a global level, they should, for a comparison across countries, they shouldn’t.¹

¹This is one of the reasons that the SKOS matching relations are not transitive.
²See https://grid.ac/
³See http://www.orgref.org/web/download.htm
text, same author) or different (different printing edition)? In all cases, the choice depends upon the application at hand.

The core question that we address in this paper is: how can we facilitate users in constructing, sharing and using context-sensitive equality relationships between semantic web datasets?

We extend earlier work on linksets [2], link scripts [18] and scientific lenses [5] by making explicit both the links and the methods for constructing those links as well as the justifications for them. We can then manipulate such explicit representations (query them, combine them, validate them) and in this way construct different equality relations depending on context.

Our approach consists of two steps: we first generate (multiple sets of) candidate correspondences (the linksets), and we explicitly represent such candidate sets, including the reasons for why they were generated as candidates and how they were generated (and by whom, and when, etc.). We then can use this rich metadata to select candidate links and combine candidate sets into context-specific lenticular lenses, which then serve as a context-specific equality relation for a view over the integrated data.4

This paper is structured as follows. We discuss related work in section 2, give formal definitions for our approach in section 3 and provide an RDF model and implementation in section 4. Section 5 introduces use cases from the STI domain, and evaluates how lenticular lenses help users to answer research questions that rely on alternative mappings across datasets. Section 6 concludes.

2 RELATED WORK

Our discussion of related work concerns three aspects: methods for identifying and generating equality relations between entities across datasets, and methods for expressing the correspondences and their metadata that allows us to select the right links on demand in such a way that they do not lead to inconsistencies.

Basic terminology. We briefly clarify some of the terminology used, and then discuss how the related work compares to the approach presented in this paper. A relation between two entities originating from different datasets is either called an RDF link [2] or more specifically a correspondence triple [7] when the link expresses an equality relation. An alignment between two datasets is called a linkset [2] when the correspondence triples that make the alignment all use the same link predicate. This predicate is called the linktype of the linkset, e.g. owl:sameAs, skos:exactmatch.

Identifying and generating correspondences. In [4], a theory is proposed for a contextualised owl:sameAs semantics that bases equality explicitly on the commitment to a set of identity criteria (property value pairs) that resources have in common, and that set can be varied depending on context. Our approach generalises [4] as we allow for other methods to construct context-sensitive equality relations beyond the simple intersection of equal property-value pairs. Furthermore, our approach allows to establish different equality criteria between different pairs of datasets, while the theory in [4] assumes a single universe across which the equality criteria apply globally (details in section 3).

To find equality relations between resources, we can make use of several tools that fall into two categories: those operating as a 'black box' (e.g. using machine learning or other heuristics) and those that rely on users for guiding the mapping process. Examples of the former are AGDISTIS [16], LogMap [14], OTO [6], and other string or graph-based similarity algorithms. These systems are intended to be generic in the sense that they do not take context or domain specific considerations into account. This makes them less suitable for our purposes: as argued above, the appropriateness of a linkset is most often determined by contextual factors.

Tools of the user-guided type seem more promising: they generate mapping triples based on user defined rules that serve as explicit representations of the context-specific identity criteria. For example, Linkage Query Writer (LQW) [11, 12] uses requests expressed in the LinQL language to discover a variety of relations between resources in one or more datasets. Similarly, SILK [18] is driven by a link specification language SILK-LSL that can be used to express the context-specific user-defined conditions under which two resources are linked. LQW and SILK-LSL support syntactic, semantic and hybrid similarity metrics. The SILK workbench UI allows users to inspect the confidence value of the links it discovers. Unfortunately, these tools do not record the justifications of their context-specific mappings in a declarative form, e.g. as RDF metadata. As a result, it is hard to decide if a particular context-specific mapping can be re-used for another purpose. The only justification of the mappings from those tools is the implicit encoding of the context in the form of the mapping rules.

Amalgame [17] is an interactive tool primarily focused on the alignment of SKOS vocabularies.5 It can produce large provenance records that capture how correspondences were created, using a combination of the PROV-O vocabulary and reified RDF triples (statements).6 However, such a reified encoding is not a suitable representation for on demand context-specific selection of candidate linksets [15].

A more pragmatic encoding is taken in the OpenPHACTS project7 which provides access to integrated biomedical Linked Data. Because users have different requirements for ‘sameness’, this access is through so-called ‘Scientific Lenses’ [5] that enable or disable specific linksets. In OpenPhacts, these linksets are represented as named graphs containing pairs of resources that are linked using a single predicate. To facilitate the selection of lenses, they are expressed as RDF using an extension of the VOID vocabulary for datasets8 that includes a reference to the justification for the linkset: i.e. the property on the basis of which the two resources were linked. Thus, the OpenPHACTS approach improves over SILK in having a declarative representation and improves over Amalgame by having a practically useful encoding. We therefore choose to build on the OpenPhacts approach.

However, OpenPhacts scientific lenses are limited in three respects. First, the model only allows to express metadata (such as confidence) at the level of the lens or linkset as a whole (the named graph). Secondly, a lens can only combine multiple linksets through taking the union of such linksets. To overcome the first limitation in our approach we express metadata at the graph and at a triple level. This allows for more fine-grained reuse of candidate links from a

---

4 A Lenticular Lens is an array of magnifying lenses, designed so that when viewed from slightly different angles, different images are magnified, from http://en.wikipedia.org/wiki/Lenticular_lens.


6 See http://www.w3.org/TR/void-

7 See http://openphacts.org

8 See https://www.w3.org/TR/void/
In this section we will define the notions of contextualised equality, allowing for union, intersection, difference, and composition (see sections 3 and 4). The third limitation of scientific lenses in Open-Phacts is that the correspondences are expressed using the linking predicate (e.g. owl:sameAs) directly. As a result, multiple competing lenses potentially introduce inconsistencies into a knowledge base. To avoid this, we introduce a unique linking predicate per individual correspondence.

**Encoding correspondences.** We must ensure that we can decorate each individual correspondence (so that they can be selected and re-used), and ensure that the semantics of alternative linksets do not interact. In [7], it is suggested to use n-ary relations to express correspondences when multiple alternative matches exist at the same time. Essentially this pattern is similar to RDF reification, which introduces a new resource of type rdf:Statement and three predicates to identify the rdf:subject, rdf:predicate, and rdf:object of the reified triple. The idea is then to associate the metadata to the newly added statement. Unfortunately, the RDF semantics does not allow us to infer the existence of the reified triple from the existence of an rdf:Statement. Furthermore, both reification and n-ary relations not only introduce an overhead of at least four new triples, but also querying becomes more difficult.

Alternatively, one can use a named graph for each triple. However, this has the same effect on query complexity, and has an additional negative effect because triple store indices are typically optimized for fewer named graphs. One way to address this shortcoming is to use an in-line syntax for expressing reified triples in data and queries ([10], see Listing 1). However, this requires RDF* and SPARQL*.

### Listing 1: In-line reification, a Statement-Level Metadata

Nguyen et al. [15] propose ‘singleton properties’, where the predicate of every individual triple (e.g. in a linkset) is uniquely identifiable, and is an ‘rdf:singletonPropertyOf’ a more generic property. This allows us to assign metadata for each candidate link to its singleton property, while at the same time, the semantics of these links do not interact across linksets. As with reification, query complexity and performance are affected. But compared to other approaches, experiments suggest that this is less affected [13].

This section discussed entity matching tools, the use of linksets and lenses to allow for different views over data, and ways to approach, experiments suggest that this is less affected [13]. But compared to other approaches, experiments suggest that this is less affected [13].

### 3 DEFINING LENTICULAR LENSES

In this section we will define the notions of contextualised equality and the supporting notions of linkset and lens, before presenting in the next sections an RDF model and an implementation for these.

The semantics of the OWL constructions for equality (owl:sameAs and owl:sameIndividual) adheres to Leibniz’ Law on identity of indiscernibles, which in quasi-RDF notation would read:

\[ \forall p : x = y \leftrightarrow (\langle x, p, v \rangle \leftrightarrow \langle y, p, v \rangle) \]  

(1)

where \( \langle \cdot, \cdot, \cdot \rangle \) denotes a triple. This bi-implication captures two separate principles, each of which is too strong for useful deployment on the semantics web. The \( \rightarrow \) direction captures the indiscernibility of identicals (identical objects share all their properties). Even the simple examples from our Introduction make clear that this is too strong, due to the quantification over all possible properties \( p \), whereas in practice, the set of predicates that is used to determine equality differs between contexts. This is the problem of context independence. The \( \leftrightarrow \) direction captures the identity of indiscernibles. This is too strong for use on the semantic web because of the open world assumption. Again, this could be repaired by restricting the set of predicates \( p \) to a finite set, such as those predicates occurring in some knowledge graph, or some ontology.

**Context-sensitive indiscernibility.** The unsuitability of the Leibniz principle for the Semantic Web was also noted by others. [4] defines a context as the set of predicates \( \Pi \) which are necessary and sufficient to determine indiscernibility and hence equality:

\[ \forall p \in \Pi : x \equiv_{\Pi} y \leftrightarrow (\langle x, p, v \rangle \leftrightarrow \langle y, p, v \rangle) \]  

(2)

Because of the restriction to predicates in \( \Pi \), \( \rightarrow \) is now context sensitive, and \( \leftrightarrow \) is limited to a closed world.

However, [4] is unclear about the treatment of properties \( p \notin \Pi \). Should these properties propagate over equality, ie.

\[ \forall \varphi \notin \Pi : x \equiv_{\Pi} y \wedge (\langle x, p, v \rangle \rightarrow (y, p, v)) \]  

(3)

or not? A simple example shows that the answer again differs per context. Drugbank [19] publishes data about more than 8000 drugs with over 200 properties per drug. Among these are the chemical structure of the drug, its biological targets, and the brand names under which it is sold. In a pharmaceutical setting, two drugs are equal if they have the same chemical structure, hence the context-defining property would be: \( \Pi = \{\text{structure}\} \). But the two properties \textit{target} and \textit{brand}, both not in \( \Pi \) should behave differently with respect to (3). Two drugs with the same structure do address the same targets, hence (3) should apply to \textit{target}; but two drugs with the same structure do not necessarily have the same brand name, hence (3) should not apply to \textit{brand}.

For these reasons, we propose a richer definition of context, which recognises that both \textit{structure} and \textit{target} are important in the pharmacological context (as opposed to the pharmacologically irrelevant property \textit{brand}), but that they play different epistemological roles: \textit{structure} determines the indiscernibility equivalence classes, and inside those classes the value of \textit{target} propagates, but it does not determine indiscernibility.

**Definition of context.** A context is defined by two sets of properties, \( \Pi \) and \( \Psi \), \( \Pi \) for indiscernibility, and \( \Psi \) for propagation:

\[ \forall p \in \Pi : x \equiv_{(\Pi,\Psi)} y \leftrightarrow (\langle x, p, v \rangle \leftrightarrow \langle y, p, v \rangle) \]  

(4)

\[ \forall p \in \Psi : x \equiv_{(\Pi,\Psi)} y \rightarrow (\langle x, p, v \rangle \rightarrow \langle y, p, v \rangle) \]  

(5)

\[ \forall p \notin \Pi \cup \Psi : \text{values of } p \text{ remain unchanged} \]  

where (4) is the definition of contextualised equality, and (5) defines contextualised propagation. This allows for drugs that are indiscernible in the pharmacological context (\( \Pi = \{\text{structure}\} \), \( \Psi = \{\text{structure, target}\} \)) to still have different values for properties outside \( \Pi \cup \Psi \), such as \textit{brand}. 

\[ \forall p \in \Pi : x \equiv_{(\Pi,\Psi)} y \leftrightarrow (\langle x, p, v \rangle \leftrightarrow \langle y, p, v \rangle) \]  

(1)

where (\( x, y, p \)) denotes a triple. This bi-implication captures two separate principles, each of which is too strong for useful deployment on the semantics web. The (\( \rightarrow \)) direction captures the indiscernibility of identicals (identical objects share all their properties). Even the simple examples from our Introduction make clear that this is too strong, due to the quantification over all possible properties (\( p \)), whereas in practice, the set of predicates that is used to determine equality differs between contexts. This is the problem of context independence. The (\( \leftrightarrow \)) direction captures the identity of indiscernibles. This is too strong for use on the semantic web because of the open world assumption. Again, this could be repaired by restricting the set of predicates (\( p \)) to a finite set, such as those predicates occurring in some knowledge graph, or some ontology.

**Context-sensitive indiscernibility.** The unsuitability of the Leibniz principle for the Semantic Web was also noted by others. [4] defines a context as the set of predicates (\( \Pi \)) which are necessary and sufficient to determine indiscernibility and hence equality:

\[ \forall p \in \Pi : x \equiv_{\Pi} y \leftrightarrow (\langle x, p, v \rangle \leftrightarrow \langle y, p, v \rangle) \]  

(2)

Because of the restriction to predicates in (\( \Pi \)), (\( \rightarrow \)) is now context sensitive, and (\( \leftrightarrow \)) is limited to a closed world.

However, [4] is unclear about the treatment of properties (\( p \notin \Pi \)). Should these properties propagate over equality, ie.

\[ \forall \varphi \notin \Pi : x \equiv_{\Pi} y \wedge (\langle x, p, v \rangle \rightarrow (y, p, v)) \]  

(3)

or not? A simple example shows that the answer again differs per context. Drugbank [19] publishes data about more than 8000 drugs with over 200 properties per drug. Among these are the chemical structure of the drug, its biological targets, and the brand names under which it is sold. In a pharmaceutical setting, two drugs are equal if they have the same chemical structure, hence the context-defining property would be: (\( \Pi = \{\text{structure}\} \)). But the two properties (\textit{target} and \textit{brand}), both not in (\( \Pi \)) should behave differently with respect to (3). Two drugs with the same structure do address the same targets, hence (3) should apply to \textit{target}; but two drugs with the same structure do not necessarily have the same brand name, hence (3) should not apply to \textit{brand}.

For these reasons, we propose a richer definition of context, which recognises that both (\textit{structure} and \textit{target}) are important in the pharmacological context (as opposed to the pharmacologically irrelevant property (\textit{brand})), but that they play different epistemological roles: (\textit{structure}) determines the indiscernibility equivalence classes, and inside those classes the value of (\textit{target}) propagates, but it does not determine indiscernibility.

**Definition of context.** A context is defined by two sets of properties, (\( \Pi \)) and (\( \Psi \)), (\( \Pi \)) for indiscernibility, and (\( \Psi \)) for propagation:

\[ \forall p \in \Pi : x \equiv_{(\Pi,\Psi)} y \leftrightarrow (\langle x, p, v \rangle \leftrightarrow \langle y, p, v \rangle) \]  

(4)

\[ \forall p \in \Psi : x \equiv_{(\Pi,\Psi)} y \rightarrow (\langle x, p, v \rangle \rightarrow \langle y, p, v \rangle) \]  

(5)

\[ \forall p \notin \Pi \cup \Psi : \text{values of } p \text{ remain unchanged} \]  

where (4) is the definition of contextualised equality, and (5) defines contextualised propagation. This allows for drugs that are indiscernible in the pharmacological context (\( \Pi = \{\text{structure}\} \)), (\( \Psi = \{\text{structure, target}\} \)) to still have different values for properties outside (\( \Pi \cup \Psi \)), such as (\textit{brand}).
Across multiple datasets All the above assumes that all properties \( p \), entities \( x \) and values \( v \) live in a single namespace. In practice, \texttt{owl:sameAs} links are often used to create links between multiple datasets, with different namespaces. So simply stating \((x, p, v) \leftrightarrow (y, p, v)\) is not enough if both sides live in different namespaces. We extend the definition above to take this into account. We use the symbol \( \equiv \) to indicate an alignment between two terms from different datasets. We choose the ”approximate” symbol because such alignments are often indeed approximate string-matching (e.g. string-matching ”New York” with ”New York”), but can also be more elaborate (e.g. dictionary-based matching ”Den Haag” with ”s Gravenhage”), or linguistic (e.g. translating ”pneumonia” to ”longontsteking”). Now a context consists not only of sets of predicates for indiscernibility and propagation, but also of an alignment procedure, i.e. a context is now \((\Pi, \Psi, \approx)\). Then the definition of contextualised equality (4) becomes

\[
\forall x, y \in \Pi \forall p, v \in \Psi : x \equiv y \Leftrightarrow \exists i : (x, i) \equiv (y, i)
\]

and similar for the definition of contextualised propagation (5).

We now have everything in place to define our central notions of linkset and lenticular lens:

Linkset. Given two datasets \( D_1 \) and \( D_2 \) and a context \((\Pi, \Psi, \approx)\), a linkset \( L \) is the set of all pairs \((x, y)\) from \( D_1 \times D_2 \) that are indiscernible in that context: \( L = \{(x, y) \in D_1 \times D_2 \mid x \equiv y\} \). In other words, a linkset is the set of all context-specific correspondences between two datasets.

Lenticular Lens. A lenticular lens is also a set of context-specific correspondences between two datasets, but is constructed from linksets through union, intersection, difference and composition:

\[
\begin{align*}
(x, y) &\in \bigcup L_i \Leftrightarrow \exists i : (x, y) \in L_i \\
(x, y) &\in \bigcap L_i \Leftrightarrow \forall i : (x, y) \in L_i \\
(x, y) &\in L_a - L_b \Leftrightarrow (x, y) \in L_a \land (x, y) \notin L_b \\
(x, y) &\in L_a \circ L_b \Leftrightarrow \exists z : (x, z) \in L_a \land (z, y) \in L_b
\end{align*}
\]

4 RDF MODEL AND IMPLEMENTATION

This section presents an RDF model for Lenticular Lenses that allows us to generate, select and combine alignments in a context-specific manner, aiming to maximise potential for future reuse. To achieve this, we combine metadata about individual correspondences using singleton properties with metadata about an entire linkset. The singleton properties are organised in a property hierarchy that allows us to capture context-specific notions of equality at different levels: moving hierarchically from linktypes for very specific tasks to linktypes for more generic tasks, and vice versa; and finding a common shared notion of indiscernibility whenever different linksets or lenses are combined for a specific task.

Example. We illustrate our model (see Figure 1) in a simple example from the STI domain. We have three linksets, \( R \), \( B \), and \( Y \) that each use different identity criteria to specify correspondences between two datasets: ETER and GRID. To generate \( R \), the entities from both datasets (research organisations) were aligned using their resource identifier. For \( B \), we applied an edit distance algorithm over organisation names, and only included pairs with similarity measure \( \theta > 0.8 \). Finally, \( Y \) refines \( B \) by further demanding that organisations must have the same country as well as \( \theta > 0.8 \).
We can use the above representation of linksets to construct a Lenticular Lens that reflects the use case specific identity criteria. In [5], Lens definitions are used to look up the linksets that should be ‘enabled’ for a specific SPARQL query: they are the named graph(s) that scope the query. Our approach differs in two important ways. First, in [5], enabling multiple linksets for a query makes all of its correspondences true for that query. In other words: the lens can only express a union of linksets. Secondly, our linksets are expressed using singleton properties, rather than the actual equality relation. We thus need to define how a lens combines linksets in more expressive ways; and how the lens can make the equality relation visible.

For example, consider the scenario where a user only wants to take into account correspondences that have a similar name (θ > 0.8), have the same resource identifier, and are in the same country. In Listing 6 we specify such a Lens as the intersection of the the linksets R and Y from the example above (expressed using the ll:operator property).

### Listing 5: Defining metadata for linkset B

#### 4.2 Using Linksets in a Lens

We can use the above representation of linksets to construct a Lenticular Lens that reflects the use case specific identity criteria. In [5], Lens definitions are used to look up the linksets that should be ‘enabled’ for a specific SPARQL query: they are the named graph(s) that scope the query. Our approach differs in two important ways. First, in [5], enabling multiple linksets for a query makes all of its correspondences true for that query. In other words: the lens can only express a union of linksets. Secondly, our linksets are expressed using singleton properties, rather than the actual equality relation. We thus need to define how a lens combines linksets in more expressive ways; and how the lens can make the equality relation visible.

For example, consider the scenario where a user only wants to take into account correspondences that have a similar name (θ > 0.8), have the same resource identifier, and are in the same country. In Listing 6 we specify such a Lens as the intersection of the the linksets R and Y from the example above (expressed using the ll:operator property).

### Listing 6: Defining metadata for Lens_R_Y.

As singleton properties do not carry semantics by themselves, we need to define how they will be used when answering queries over the datasets to which they apply. This is where the linktype hierarchy comes into play: merging linksets where the linktypes are normalized to a shared ancestor in the property hierarchy. Using the specification in Listing 6 we generate a SPARQL query (Listing 7) that does just this: where the pairs of resources from linksets R and Y are the same, we insert an equality relation using the most specific common ancestor for the respective singleton properties. The query is in essence a template that can be applied to any combination of two linksets that are merged using the intersection operator.10 The resulting lens :lens_R_Y can now be used as in [5].

#### Listing 7: SPARQL query for generating Lens_R_Y without metadata implementing an intersection over linksets.

Although the lens described in Listing 7 is very simple, if there is a need to combine multiple alignments, finding the common shared linktype becomes more challenging. We provide formal definitions of supported operators (union, intersection, difference and composition) in section 4.3.

### 4.3 Complex Operations over Linksets

A lens is a combination of linksets and/or other lenses using set-like operators such as union, intersection, difference and composition. Although these operators are implemented in SPARQL, they can not be used off-the-shelf in our framework for three reasons. First, the equality relation is symmetric: the direction of a triple is irrelevant for the indiscriminability relationship: (e1, r, e2) iff (e2, r, e1) (where r is the linktype). Second, in the proposed approach, the identity is represented through a unique singleton property: two instances of the same identity between e1 and e2 will look like (e2, r, e1) and (e3, r, e1) respectively, with r different from r'. Third, new singleton properties will have to be generated for the resulting lens. Other than these technicalities, the operators behave as their set-theoretic counterparts. These issues are addressed in three steps.

#### Algorithm 1: Algorithm for generating a Lens using the UNION operator.

An alignment is composed of three graphs: main (the correspondences graph), generic (the generic metadata graph) and specific (the singleton graph).
We first present two illustrative use-cases in subsection 5.1. In As usual with methodological proposals, it is not obvious how (approximate string matching, i.e. the procedure (Institutions, Organisation), the datasets selecting a context location. aligns them on the basis of matching both their name and their independently of their geographic locations, while case study 2 consequently, case study 1 needs to align firms using the company name, the system associates a data-linking task to the context of a specific case.
with this complex case. The aim is to investigate whether the university ranks in the Leiden Ranking dataset can be predicted from the properties of the university and its environment in other datasets. For this, we need to align universities across five datasets: ETER (providing characteristics of European higher education institutions), GRID (a worldwide collection of academic research institutes), Leiden Ranking (performance metrics of over 800 major universities worldwide), GRID_Enr and ETER_Enr. The last two datasets are subsets of GRID and ETER enriched with geographic boundaries from GADM\(^1\) and a count on the number of worldwide organisations (GRID_Enr) and European universities (ETER_Enr). The dotted lines in fig. 3 show the complex mappings between these datasets that are needed to answer the research question. Because of this complexity, this use case is suitable to test the strength of our approach.

**Linkset creation.** Listing 8 gives a textual representation of what was shown in the user interface of Figure 2: research question, datasets, types of entities to be matched, properties to be aligned, and the matching mechanism to be used. ETER is directly aligned to three different datasets using different types, properties and mechanisms, resulting in \( \text{Linkset}_1 \) through \( \text{Linkset}_3 \).

**Research Question**

Can we predict the CWTS scores from characteristics of the university?

**Mapping (Dataset|EntityType)**

- Leiden Ranking | University
- GRID | Institution
- GRID | GRID_GADM_enriched | Institution
- ETER | University
- ETER | ETER_GADM_enriched | University

**Linkset** specifications

- \( \text{Linkset}_1 \) specifications
- \( \text{Linkset}_2 \) specifications
- \( \text{Linkset}_3 \) specifications

**Dataset** (ETER, GRID)

- (ETER, GRID)
- (ETER, GRID)
- (ETER, Leiden Ranking)
- (ETER, Leiden Ranking)
- (University, University)
- (University, University)
- (inst_name, name)
- (eng_inst_name, name)
- (inst_name, name)
- (eng_inst_name, name)

**Mechanism** Approximate Similarity

- Approximate Similarity
- Approximate Similarity
- Approximate Similarity

**Entity Type** (University, Institution)

- (University, Institution)
- (University, Institution)
- (University, University)
- (University, University)

**IndisProp** (inst_name, name)

- (eng_inst_name, name)
- (inst_name, name)
- (eng_inst_name, name)
- (inst_name, name)

**Property Selection**

- \( \text{PropertySelection}(\Psi) \)
- \( \text{PropertySelection}(\Psi) \)
- \( \text{PropertySelection}(\Psi) \)

**Final Integration Lens**

- \( \text{FinalIntegrationLens} \)

**Correspondence Filter**

- \( \text{CrF}_1 = \text{FILTER}(\text{STRENGTH}=2, \text{THRESHOLD} \geq 8.5, \text{ACCEPT}) \)
- \( \text{CrF}_2 = \text{FILTER}(\text{STRENGTH}=2, \text{THRESHOLD} \geq 8.5) \)

**Property Selection**

- \( \text{PropertySelection}(\Psi) \)

**Final Integration Lens**

- \( \text{FinalIntegrationLens} \)

**Listing 8: \((\Xi, \Psi, \approx)\) of case study 3**

**Lens creation.** From the linksets generated in the previous step, we generate lenses \( \text{Lens}_1 = \text{Linkset}_1 \cup \text{Linkset}_2 \) and \( \text{Lens}_2 = \text{Linksets} \cup \text{Linkset}_3 \) to align ETER with GRID and the Leiden Ranking. This is also captured by the solid arrows in Figure 3.

**View and Final Integration Lens.** As explained above, constructing a View is the final step for expressing the user’s context-specific integration perspective over the data (see Figure 3). The system allows the construction of complex combinations of lenses. Listing 9 shows the specification for generating the final View required for use case 3, where \( \text{Lens}_3 \) represents the final contextual view and Final Integration Lens.

**Listing 9:** Specifications for generating a final lens and view

- \( \text{Lens}_3 \)

**Integration:**

\[
\text{Lens}_3 = \text{Linkset}_1 \cap \text{Linkset}_2 \cap \text{Linkset}_3 = \text{Linkset}_1 \cap \text{Linkset}_2 \cap \text{Linkset}_3 = (\text{Linkset}_1 \cup \text{Linkset}_2) \cap (\text{Linkset}_3 \cup \text{Linkset}_4)
\]

\( \text{Lens}_3 \) specifies different filters that refine the conditions on which a correspondence is to be included. For example, \( \text{CrF}_1 \) is a filter that includes only those correspondences from \( \text{Lens}_3 \) that have \( \text{STRENGTH}=2 \), that have a \( \text{THRESHOLD} \) greater or equal to 8.5, and that have been evaluated as \( \text{ACCEPTED} \). Listing 9 also shows a list of the 15 properties (for \( \Psi \) that are used to construct the View table; they are the ‘variables’ for which the data is studied.

**5.3 Complexity**

**Time complexity.** Given that creating a linkset requires two datasets and \( X \) and \( Y \), its time complexity depends on the alignment algorithm used and the sizes of the datasets. For example, using a naive string-based matching algorithm, each entry of the source dataset is compared to all those of the target dataset which takes \( O(|X||Y|) \) time. Instead, using a blocking solution (implemented in the framework) takes \( O(|X||Z|) \), where \( Z \subseteq Y \) with \( |Z| \ll |Y| \). Furthermore, the cost.
for decorating the generated correspondences is low: an insertion of cost $O(1)$ for each correspondence. Other performance parameters are strong triple store dependent. Computing an exact similarity between resources is fully implemented in SPARQL, and in practice takes less than a minute to create a linkset between GRID (74523 instances) and OrgRef (32010 instances) on commodity hardware.

**Space Complexity.** The space complexity is determined by the number of matches found between the datasets (the size $|L|$ of a linkset), with $0 < |L| < |X||Y|$. However, in practice $|L| \ll \max(|X|, |Y|)$. The decoration adds only a fixed number $n$ of triples per correspondence, resulting in an output $n$ times bigger than a non-decorated linkset.

## 6 CONCLUSION

This paper presents a system for constructing context-specific links between data-sets: Lenticular Lenses. A context-specific approach is needed because whether or not resources in different datasets are to be considered indiscernible strongly depends on the purpose and the task for which the datasets are to be used. We do not propose any new disambiguation method. Instead, our system allows the use of existing alignment methods to construct a linkset of potential mappings while annotating these mappings with rich meta-data using the proposed meta-model. Such meta-data enables Linked Data users to combine a variety of matching tools to obtain multiple context-sensitive alignments through simple SPARQL queries over the correspondences and their annotations. Compared to owl:sameAs, the proposed context:sameAs provides generic/shared metadata for alignment reproducibility, and specific correspondence metadata for context-specific re-usability and validation.

In three different case studies from the social science discipline of science and technology studies, we showed that our approach provides the necessary functionality to fulfill the requirements of complex realistic alignment problems: we maintain a rich meta-model that allows the user to select candidate alignments on a variety of properties such as tool of origin, alignment strength, and composition; we maintain a full declarative provenance trail and re-usability. Our three realistic case studies from social science have shown that all of these features are indeed required to develop an environment for the construction and manipulation of context-sensitive links between datasets.

As singleton properties may not be easily used, the system allows for converting the enriched alignments into the usual flat format. It also allows for importing flat alignments generated by other tools. Although these alignments come without annotation, it is still possible to document the correspondences (e.g. validation). This can serve as a justification for selecting contextually valid correspondences. As future work, we plan to integrate state of the art alignment algorithms such as [1] and automate the conversion of other refiliation models into singleton properties and vice versa. Another important feature to include is the detection of a chain of equality predicates across datasets and there respective contextual information. Furthermore, referenced resources evolve through time. This leaves us with the need to investigate its impact over the network of correspondences in the Lenticular Lens system, including changes of statements due to new dataset versions. Additionally, we plan to enable the linking of datasets under construction to external datasets. Finally, beside deploying and validating a production version of the user interface, we plan to investigate an ontology for the Lenticular Lens (II) vocabulary.

## REFERENCES


