Web Of Data: Data Linking and Link Validation

Pablo Mollá Chárlez

Contents

1 Introduction

In our earlier discussions, we explored the foundations of ontology alignment, addressing challenges like conceptual heterogeneity, internal and external structure similarities, and measures such as Wu and Palmer to quantify semantic relationships. These tools and concepts laid the groundwork for understanding how ontologies interact and align.

From this point forward, we will shift our focus to data linking procedures and techniques, which are critical for connecting disparate datasets and establishing relationships between entities across different systems or ontologies.

2 Data Linking

2.1 Formal Definition

Data linking, also known as **identity link detection**, involves identifying whether two descriptions of entities refer to the same real-world entity (e.g., the same person, book, or gene). This process is essential for integrating datasets and achieving semantic interoperability. The formal definition is as follows:

- **Input**: Two sets of resources U_1 and U_2.
- **Goal**: Partition $U_1 \times U_2$ into:
  - S = $\{(u1, u2) \in U_1 \times U_2 :$ owl:sameAs(s,t)$\}$ ~ Pairs (u_1, u_2) where u_1 and u_2 are the same entity (e.g., owl:sameAs).
  - D = $\{(u1, u2) \in U_1 \times U_2 :$ owl:differentFrom(s,t)$\}$ ~ Pairs (u_1, u_2) where u_1 and u_2 are different entities (e.g., owl:differentFrom).

Depending on how the partition is done, we can obtain a Total Method (completes the partition such that $S \cup D = U_1 \times U_2$) or a Partial Method (only identifies a subset, $S \cup D \subset U_1 \times U_2$). The problem is that, usually data linking is more complex for graphs (e.g., Semantic Web data) compared to traditional tables (e.g., relational databases):

	Databases	Semantic Web
Schema/Ontologies	Same schema	Possibly different ontologies in the same dataset
Multiple types	Single relation	Several classes
Open World Assumption	NO	YES
UNA-Unique Name Assumption	Yes	May be no
Data volume	XX Thousands	XX Millions/Billions (e.g., DBpedia has 1.5 billion triples)
Multiple values for a property	NO	YES P1 hasAuthor "Michel Chein" P1 hasAuthor "Marie-Christine Rousset"

Figure 1: Data Linking: Tables vs. Graphs

2.2 Matching Approaches for Data Linking

1. **Manual Description Matching**: Relies on human experts to match entities based on their descriptions or metadata. **Strengths:** Useful for creating a reference or first sample. **Weaknesses:** Does not scale for large datasets. **Example:** Crowdsourcing to match entries in a dataset.

2. **URI Matching**: Compares the Uniform Resource Identifiers (URIs) of resources to check if they refer to the same entity. **Strengths:** Straightforward for datasets with consistent identifiers. **Weaknesses:** Limited to datasets that use URIs. **Example:** Matching "http://rdf.insee.fr/geo/regions-2011.rdf#REG11" to a similar URI in Eurostat.

3. **ID Matching**: Matches entities based on unique identifiers, such as database IDs or other reference numbers (ISBN, SSN, or DOI). **Strengths:** Reliable when IDs are unique and consistent. **Weaknesses:** IDs are often local to the database and may require normalization. **Example:** Matching books by ISBN numbers.

4. **Context-Based Matching**: Uses the surrounding context of entities (e.g., relationships or usage patterns) to determine if they represent the same real-world entity. Basically, it projects data onto an external resource (e.g., DBpedia, Geonames) and assesses relationships. **Strengths:** Incorporates external knowledge for linking. **Weaknesses:** Requires external resources and alignment. **Example:** Using DBpedia to enrich terms with external relations.

5. **Content-Based Matching**: Compares the actual content (e.g., text or data) associated with the entities to assess similarity. **Strengths:** Works well with unstructured datasets. **Weaknesses:** Requires robust similarity measures and preprocessing. We can distinguish two approaches:

   - **Bag of Text:** Compares raw text content.
   - **Structured Similarity:** Analyzes structured attributes (e.g., names, descriptions).

   Example: Matching similar product descriptions in e-commerce datasets.

6. **Term-Based Matching**: Focuses on matching terms or keywords used to describe entities, often using natural language processing or ontological concepts. It relies on linguistic normalization and similarity measures. **Strengths:** Handles variations in terms effectively. **Weaknesses:** Sensitive to linguistic ambiguities and translations. Tools include:

   - Normalizers (e.g., Stemmers, Tokenizers).
   - Linguistic resources (e.g., WordNet).
   - Similarity measures (e.g., cosine similarity, n-grams).

   Example: Matching "University of Paris" with "Paris University."

7. **Structure-Based Matching**: Compares the structural relationships between entities, such as **class hierarchies** or **properties**, to assess equivalency. **Strengths:** Explores graph topology to infer links. **Weaknesses:** Computationally expensive and less scalable. Techniques include:

   - Graph matching.
   - Learning weights for properties.

   Example: Matching social network graphs based on node similarity and edge relationships.

8. **Cross-Lingual RDF Data Linking**: Links entities that represent the same real-world object but are described in different languages, utilizing multilingual data sources.

Each matching approach has specific strengths and weaknesses depending on the dataset context, structure, and available resources. For large-scale data linking, automated methods like URI, ID, and content-based matching are preferred, while manual and context-based methods are used for specialized or initial mappings.

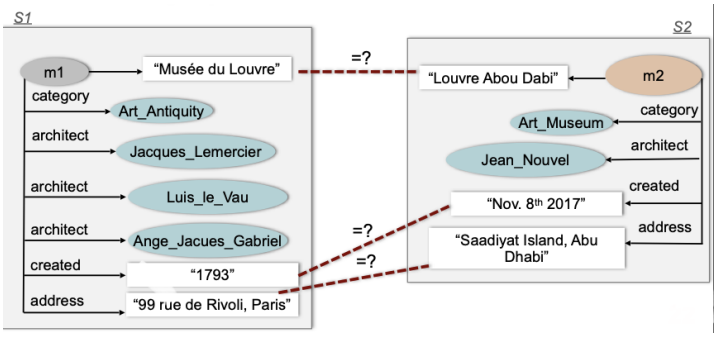

Figure 2: Instance-Based Approach

2.3 Data Linking Approaches

Let's explain each of the four data linking approaches:

- **Instance-Based Approaches**: These approaches focus solely on data type properties (attributes). They compare the values of attributes (e.g., a person's name or age) between different instances and determine whether they represent the same entity based on the similarity of these values.
- **Graph-Based Approaches**: In addition to data type properties (attributes), graph-based approaches also take into account object properties (relations) between entities. These approaches use the **relationships between entities** (e.g., "Author of", "Part of") to propagate similarity scores and make collective linking decisions. This allows for more complex similarity assessments, considering the structure of the graph.

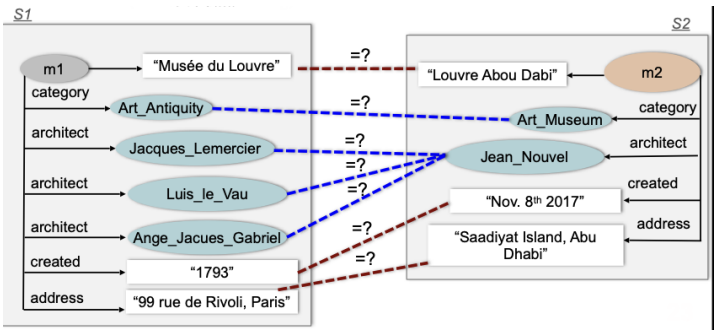

Figure 3: Graph-Based Approach

- **Supervised Approaches**: These methods require manual effort to create labeled training data. Experts provide pairs of entities that are either linked or not linked, which are then used to train models to predict whether other entity pairs are linked. This often involves interactive feedback loops to refine the model's predictions.

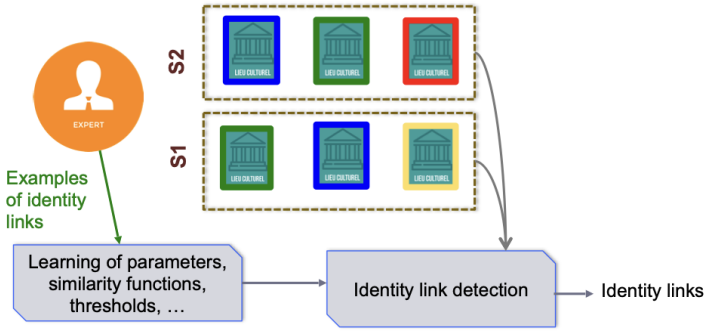

Figure 4: Supervised Approach

- **Rule-Based Approaches**: These approaches rely on explicit rules defined by experts. These rules can be encoded within the ontology or given in another format (e.g., a knowledge base), and they help determine how entities should be linked. The rules may include conditions like "if two entities have the same name and are linked by the relation 'authorOf', then they are likely the same entity."

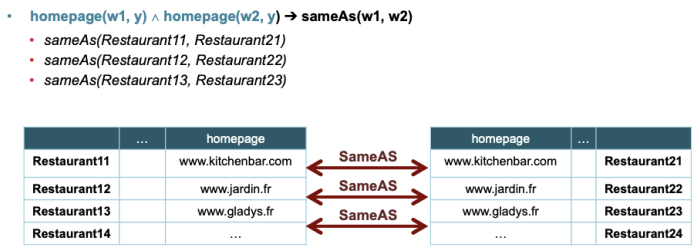

Figure 5: Rule-Based Approach

2.4 Evaluation Metrics for Data Linking

We can distinguish between evaluating effectiveness, efficiency and robustness while comparing to benchmarks:

- **Effectiveness**: Assesses the quality of linking results using:
  - **Recall**: Proportion of correctly identified links compared to all true links.

$$\text{Recall} = \frac{\#\text{Correct Links (System)}}{\#\text{Correct Links (Ground Truth)}}$$

  - **Precision**: Proportion of correctly identified links out of all links found by the system.

$$\text{Precision} = \frac{\#\text{Correct Links (System)}}{\#\text{Links (System)}}$$

  - **F1-Measure**: Harmonic mean of recall and precision.

$$\text{F1} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

- **Efficiency**: Measures the time and space required, focusing on minimizing the search space and user interactions.
- **Robustness**: Evaluates the system's ability to handle data errors and inconsistencies.
- **Benchmarking**: Uses standardized benchmarks (e.g., **OAEI**, **Lance**) to compare performance across systems.

2.5 L2NR Method

LN2R (Logical and Numerical Reference Reconciliation) is a graph-based, unsupervised, and informed method used for detecting identity links between entities in datasets. It combines logical reasoning and numerical similarity measures to establish relationships between entities while leveraging ontology axioms for consistency and precision. Key features include:

1. **Ontology-Axiom-Based Reasoning:** LN2R uses ontology axioms to infer relationships and detect inconsistencies in identity links:
   - **Disjunction Axioms** (DISJOINT(C, D)): Two classes are disjoint if an entity cannot belong to both. Formally, the logical semantics are: $\forall X,\ C(X) \Rightarrow \neg D(X)$.

- **Functional Property Axioms** (PF(P)): A property is functional if an entity can have at most one value for it. Formally, the logical semantics would be: $\forall X, Y, Z,\ P(X,Y) \wedge P(X,Z) \Rightarrow Y = Z$.
- **Inverse Functional Property Axioms** (PFI(P)): A property is inverse functional if its inverse is unique. The logical semantics are: $\forall X, Y, Z,\ P(Y,X) \wedge P(Z,X) \Rightarrow Y = Z$.

2. **Assumptions on Data:** Unique Name Assumption (UNA): It is assumed that entities from the same source have distinct identifiers (IRIs) and in terms of Local Unique Name Assumption (LUNA), certain properties (e.g., social security number) uniquely identify entities locally.

3. **Use of SWRL Rules:** Semantic Web Rule Language (SWRL) is applied to extend reasoning capabilities by generalizing functionality and inverse functionality axioms over sets of properties. For instance, for functionality: $\forall X, Y, Z,\ P_i(X,Y) \wedge P_i(X,Z) \Rightarrow Y = Z,\ \forall i \in [1..n]$ and for inverse functionality: $\forall X, Y, Z,\ P_i(Y,X) \wedge P_i(Z,X) \Rightarrow Y = Z,\ \forall i \in [1..n]$.

4. **Graph-Based Approach:** The method builds contextual graphs around entities by exploiting relationships and ontology axioms. These graphs help detect identity links and inconsistencies between entities.

The process followed with the L2NR Method is:

1. **Input Data**: Two datasets with ontologies and potential identity links.

2. **Axiom Application**: Apply disjunction, functionality, and inverse functionality axioms to infer relationships and eliminate invalid links.

3. **Reasoning**: Use logical reasoning and SWRL rules to infer new identity links (e.g., **owl:sameAs**).

4. **Validation**: Detect inconsistencies by validating inferred links against ontology axioms.

Such method can be used in the following cases:

- **Entity Matching in Knowledge Graphs:** Aligning entities from datasets like DBpedia and Wikidata.
- **Ontology Alignment:** Reconciling classes and properties in two different ontologies.
- **Data Cleaning:** Identifying and resolving erroneous identity links.

LN2R combines logical precision with the flexibility of numerical and graph-based approaches, making it robust and adaptable for complex data linking tasks.

2.6 L2R Method

L2R stands for Logical method for Reference Reconciliation. It uses resolution principles and a set of inference rules to reconcile and deduplicate references across different sources. Key concepts in L2R include:

- **Horn Clauses**: The method works with Horn clauses, which are logical expressions of the form: "$C1 \wedge C2 \vee C3$", meaning one of the clauses must be true (a logical "OR").
- **Unit Clauses**: These are clauses that are fully instantiated (i.e., they do not contain variables but specific entities).
- **Resolution Principle**: The method applies a unit resolution rule until saturation (i.e., when no further deductions can be made). The resolution rule is defined as: "$C1 : (L1),\ C2 : (L2 \vee C) \implies C1,2 : (C)$". This rule resolves two clauses "C1" and "C2" into a new clause "C1,2".
- **Inference Process**: The algorithm starts with a set of RDF facts and applies logical rules to deduce new facts. For example, if a rule indicates that two references are not reconciled (i.e., "$src1(X) \wedge src1(Y) \wedge (X \neq Y) \implies \neg Reconcile(X,Y)$"), it generates a new clause that can be used for further inference.

L2R uses an automatic generation of inference rules as follows:

1. **Translation of UNA(src1)**: This involves rules that prevent the reconciliation of two references if certain conditions are met. For instance, $src1(X) \wedge src1(Y) \wedge (X \neq Y)$ implies that X and Y cannot be reconciled ($\neg Reconcile(X, Y)$).

2. **Translation of LUNA(R)**: Similar to UNA, but applied to relationships (e.g., $R(Z, X) \wedge R(Z, Y) \wedge (X \neq Y)$), it ensures that references in relationships are not reconciled unless certain conditions are met.

3. **Translation of DISJOINT(C, D)**: If two classes (e.g., $C(X)$ and $D(Y)$) are disjoint, no reconciliation should be allowed between their instances ($\neg Reconcile(X, Y)$).

4. **Translation of PF(R)**: This rule ensures that if two references X and Y are reconciled and both are related to other entities through the same relation R, those related entities are also reconciled (i.e., $Reconcile(Z, W)$).

5. **Translation of PF(A)**: Extends PF to other types of facts like locations or names. For instance, if $Reconcile(X, Y)$ holds, and $MuseumName(X, Z)$ and $MuseumName(Y, W)$ hold, then the entities Z and W are considered synonyms ($SynVals(Z, W)$).

The L2R Algorithm presents the following properties:

- **Termination**: The algorithm terminates because it doesn't use function symbols, ensuring that the resolution will eventually stop.
- **Completeness**: The algorithm can deduce all fully instantiated unit clauses from the knowledge base, assuming the set of inference rules is correct.

Besides the previous properties, the following theorem guarantees that if the set $R \cup F$ (horn clauses and unit facts) is satisfiable, then the inferred facts (deduced through the resolution process) will be part of the set $SatUnit(R \cup F)$.

Theorem Let R be a set of un Horn clauses without functions. Let F be a set of unit clauses fully instantiated. If $R \cup F$ is satisfiable, then:

$$\forall p(a), (R \cup F \vDash p(a)) \implies p(a) \in SatUnit(R \cup F)$$

With $p(a)$, a unit clause fully instantiated and $SatUnit(R \cup F)$ is the set of inferred clauses by applying the unit resolution until saturation on $R \cup F$.

In summary, L2R uses logical resolution techniques to reconcile references between different sources of information, applying a set of predefined inference rules that account for disjointness, relationships, and synonymy.

2.7 N2R Method

The N2R method is a numerical approach for reference reconciliation, designed to compute a similarity score between pairs of references based on their common descriptions. N2R method uses a numerical approach to quantify how similar two references are by comparing their descriptions. This comparison is done through a set of common attributes and relations that both references share. The goal is to calculate a similarity score for the pair of references, which can be used for reconciliation. Key concepts include:

1. **Similarity Measures**: N2R method uses known similarity measures like **Jaccard** or **Jaro-Winkler** to quantify how similar the references are. These measures help compare the values and relations associated with the references.

2. **Coherence with L2R**: N2R method is designed to complement the L2R method (Logical Reference Reconciliation) method, taking into account the results of L2R, such as:

- $Reconcile(i, i')$: Whether two references are reconciled.
- $\neg Reconcile(i, i')$: Whether two references are not reconciled.
- $SynVals(v, v')$: Whether two values are synonyms.
- $\neg SynVals(v, v')$: Whether two values are not synonyms.

These results from L2R can influence the similarity score computed by N2R. The common description proceeds as follows: For each pair of references (i, i'), N2R method computes a common description by identifying shared attributes and relations between the two references. Here are the components:

- **Common Attributes**: The set $CAttr(i, i')$ consists of attributes shared between reference i and i'. An attribute $a(i, v)$ is common if it appears in the description of both references, with v being the value of attribute a for reference i, and similarly for reference i':

$$CAttr(i, i') = \{a \mid \exists v, v' \in Val,\ [a(i, v) \in Desc(i) \wedge a(i', v') \in Desc(i')]\ \}$$

- **Common Relations**: The set $CRel(i, i')$ includes relations between the references. A relation $r(i, j)$ is common if it appears in the description of both references:

$$CRel(i, i') = \{r \mid \exists j, j' \in I, [r(i, j) \in Desc(i) \wedge r(i', j') \in Desc(i')]\ \}$$

- **Set of Values Associated with a Reference**: $a^+(i) = \{v \mid \forall v, [a(i, v) \in Desc(i)]\ \}$: This set includes all values associated with reference i.
- **Set of References Associated with a Reference**: $r^+(i) = \{j \mid \forall j, [r(i, j) \in Desc(i)]\ \}$: This set includes all references that are related to reference i via relations.
- **Set of References to Which a Reference is Associated**: $r^-(i) = \{j \mid \forall j, [r(j, i) \in Desc(i)]\ \}$: This set includes all references to which reference i is related.

The role of N2R is to compute a numerical similarity score by analyzing the common attributes, relations, and sets of associated values and references. The method considers both the structure (the relations between entities) and the content (the attributes and their values). It provides a numerical approach to reference reconciliation, which can be used to complement N2R's logical framework by offering a more concrete similarity measure between references.

3 Link Validation

3.1 Philosophical and Operational Perspective

The main points that make identity complicated from a philosophical perspective are:

1. **Identity does not hold across modal contexts**: Identity can be affected by different possible worlds or hypothetical situations. What something is in one context may not be the same in another context, creating complexity when trying to define or compare identities across varying circumstances.
2. **Identity is context-dependent (Geach, 1967)**: The way we understand or determine identity can change depending on the specific context in which it is being considered. What counts as the "same" thing can vary with different perspectives or frames of reference.
3. **Identity over time poses problems**: This concerns how identity persists over time, especially when an object undergoes changes. For example, a ship may still be considered the same ship even after many or all of its parts are replaced. This raises questions about what makes something "the same" over time, despite alterations.

These points illustrate that identity is not a simple or static concept, but rather one that is influenced by context, changes over time, and the specific circumstances in which it is considered. From an operational perspective, identity is complicated for the following reasons:

1. **Absence of explicit identity statements**: Unless explicitly stated that two things are different, the lack of an identity statement does not necessarily mean they are not identical. This creates ambiguity in situations where identity is not clearly defined.

2. **Distinguishing between IRI and information resources**: It can be difficult to differentiate between an Internationalized Resource Identifier (IRI) referring to a non-information resource and its corresponding information resource, complicating the assignment of identity.

3. **Differences in modeler opinions**: Modelers may have different interpretations or opinions about whether two objects are the same, leading to inconsistencies in how identity is applied or understood.

4. **Imprecision in data linking**: Data linking approaches are often not 100% precise, meaning there can be errors or uncertainties when attempting to link or identify resources.

5. **Lack of well-defined and standardized identity links**: Existing identity links like $rdfs:seeAlso$ and $skos:exactMatch$ often lack formal semantics, which can lead to confusion and inconsistencies in how identity is represented and linked across systems.

These points emphasize the challenges of establishing and maintaining consistent identity in operational settings, where ambiguity, imprecision, and lack of standardization complicate the process.

3.2 How to limit the sameAs problem?

To limit the **sameAs** problem, which refers to the challenge of correctly identifying whether two different IRIs (Internationalized Resource Identifiers) refer to the same real-world entity, several strategies can be employed:

1. **Help users and applications identify IRIs referring to the same real-world entity**: Provide tools and guidelines to assist users and systems in determining whether different IRIs represent the same real-world entity or distinct entities, reducing confusion and errors. For instance:

   - **Centralized Identity Management Services**: Implement centralized services that track and manage identities, ensuring consistency in how entities are identified across different systems and applications. This can help reduce ambiguity by offering a single authoritative source for identity verification. They are designed to address the challenge of asserting identity links between multiple IRIs that refer to the same real-world entity. The hypothesis is that having multiple IRIs referring to the same entity makes asserting identity links difficult and prone to errors. The solution proposed is one globally unique and persistent identifier per real-world entity. This ensures that all IRIs referring to the same entity can be associated with a single, authoritative identifier, which makes it easier and less error-prone to establish identity links. For instance, URNs (Uniform Resource Names) provide location-independent, persistent identifiers for entities, using unique namespaces registered with IANA (Internet Assigned Numbers Authority). They ensure global uniqueness, e.g., "urn:isbn:0-7475-4624-X" for a book. Another example is OKKAM which assigns a unique ID to each real-world entity, linking it to a profile containing attributes and external IRIs (e.g., from DBpedia or Freebase) to help distinguish the entity.
   - **Identity Observatories**: Set up systems that continuously monitor and record identity assertions and links, enabling the detection of inconsistencies or errors in identity relationships and allowing for timely corrections. **Identity Observatories** aim to resolve ambiguity in the identities of real-world entities by identifying and tracking identity links (e.g., **owl:sameAs**) and calculating their transitive closure. This approach helps clarify when different names or IRIs refer to the same entity. Some examples include:

     –

- SAMEAS.ORG (Glaser et al., 2009): Uses 346M RDF triples from multiple sources to create 62.6M identity bundles by merging identity and similarity predicates. Hosted at SAMEAS.ORG, though it is no longer maintained.
- LODSYNDESIS (Mountantonakis and Tzitzikas, 2018): Based on 44M **owl:sameAs** triples, resulting in 24M equivalence classes, covering over 65M terms. Hosted at GitHub in LODSYNDESIS.
- SAMEAS.CC (Beek et al., 2018): The largest collection of **owl:sameAs** triples, covering 179M terms with 49M equivalence classes. Hosted at SAMEAS.CC.

Despite their technical limitations, identity observatories are widely used in Linked Data applications, not only to understand the meaning of IRIs but also for various practical use cases.

2. **Detect erroneous identity links / Validate correct ones**: Establish mechanisms to automatically detect incorrect identity links and validate the correctness of existing identity links. This can involve both automated and manual verification processes to ensure the accuracy of **sameAs** assertions. For instance:

   - **Inconsistency-based Approaches**: Use approaches that focus on identifying and resolving inconsistencies in identity links. This may involve conflict detection algorithms that highlight potential issues when conflicting identity links are found.
   - **Content-based Approaches**: Utilize the actual content or attributes associated with the IRIs to assess whether they refer to the same entity. This could involve comparing data points such as names, dates, or other attributes that describe the entity to help establish identity.
   - **Network-based Approaches**: Leverage the network or graph structure of linked data to identify patterns and connections between IRIs. Analyzing how entities are related within a larger context (such as through shared relationships or co-occurrence) can provide clues about whether they are likely to be the same entity.

3. **Propose alternative semantics for identity**: Develop and implement new semantics or formal definitions for identity that go beyond the traditional "sameAs" predicate. This could include creating new predicates or concepts for weak identity or similarity, which can help capture situations where entities are not identical but still closely related. For instance:

   - **Weak-Identity and Similarity predicates**: Introduce predicates like "similarTo" or "weaklyIdentical" that allow for cases where two entities may not be strictly identical but are sufficiently similar to be treated as equivalent in some contexts. This helps to distinguish between exact identity and cases of near-equality or approximation.
   - **Contextual Identity**: Recognize that the identity of an entity may depend on the context in which it is being used. By considering the context, it is possible to define identity more flexibly, allowing for different interpretations depending on the situation (e.g., an entity might be considered "the same" in one context but "different" in another).

These strategies aim to reduce the challenges of the "**sameAs**" problem by providing better mechanisms for identifying, validating, and managing entity identities in linked data systems. They offer ways to deal with ambiguity, inconsistency, and complexity, ensuring more accurate and context-sensitive identity assertions.

3.3 How to detect erroneous identity links?

To detect erroneous identity links, several types of information can be leveraged:

1. **Content**: Analyzing the actual attributes or data associated with the linked entities can help detect errors. If the content (e.g., names, dates, or other properties) of two linked entities contradicts or is inconsistent, it may indicate an erroneous identity link.

2. **Identity Network**: The structure of the relationships between entities (e.g., how they are linked to other entities) can provide clues. If two entities are linked in ways that don't make sense based on their connections (e.g., through shared relationships or common attributes), it might suggest that they are incorrectly linked.

3. **UNA (Unique Naming Assumption)**: This assumption helps by considering that each entity should only have one unique identifier. If two IRIs are linked as the same entity but are assigned different identifiers in different contexts, this inconsistency can be flagged as a potential error.

4. **Trustworthiness**: The reliability of the sources or data used to assert identity links can affect their accuracy. If a link is based on unreliable or low-quality data sources, it's more likely to be erroneous. Trustworthiness can be assessed by examining the reputation of the data source or the consistency of the link across trusted databases.

By combining these factors, erroneous identity links can be identified more effectively through inconsistencies in the content, relationships, naming assumptions, and the trustworthiness of the data sources.

3.4 Contextual Identity Links

Contextual Identity Links refer to cases where identity is not absolute but dependent on specific contexts or subsets of properties. This means that two resources (e.g., entities) can be considered identical in some aspects but not in others, depending on the context in which they are being compared. Key points include:

- **Weaker Kinds of Identity:** Identity can be defined more flexibly by considering only certain properties of the resources, rather than assuming full equivalence across all attributes. For example:
  - **Same in One Context:** Two medicines may be considered identical when comparing their chemical substance.
  - **Different in Another Context:** However, they may be considered different when comparing their price, as they might be produced by different companies.
- **Context-Dependence:** As noted by Geach (1967), identity is not always absolute but can depend on the context in which it is evaluated. This means that the criteria for determining identity can vary based on the situation.
- **Contextual Identity Relation:** Raad et al. (2017) introduced a new contextual identity relation, which helps identify the most specific contexts in which two instances (or resources) should be considered identical. The process of identifying these contexts can be guided by semantic constraints, provided by domain experts, ensuring that the context is relevant and accurate.
- **Sub-Ontology of Domain Ontology:** Contexts are defined as a sub-ontology of the broader domain ontology. This means that the broader domain knowledge is used to define specific contexts in which identity is relevant.
- **Lattice Organization:** All possible contexts are organized in a lattice structure, which uses an order relation to represent how contexts are related and prioritized. This helps in systematically determining the most appropriate context for identity comparison.

In essence, contextual identity links allow for a more nuanced and flexible understanding of identity, acknowledging that two resources may be the same in one context (e.g., chemical substance) but different in another (e.g., price). This approach allows identity to be defined in a way that is contextually appropriate, reducing ambiguity and making it easier to handle complex scenarios.