Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Not Quite the Same: Identity Constraints for the Web of Linked Data

Gerard de Melo
ICSI Berkeley
1947 Center St.
Berkeley, CA 94704, USA

Abstract

Linked Data is based on the idea that information from different sources can flexibly be connected to enable novel applications that individual datasets do not support on their own. This hinges upon the existence of links between datasets that would otherwise be isolated. The most notable form, `sameAs` links, are intended to express that two identifiers are equivalent in all respects. Unfortunately, many existing ones do not reflect such genuine identity. This study provides a novel method to analyse this phenomenon, based on a thorough theoretical analysis, as well as a novel graph-based method to resolve such issues to some extent. Our experiments on a representative Web-scale set of `sameAs` links from the Web of Data show that our method can identify and remove hundreds of thousands of constraint violations.

1 Introduction

The vision of a Web of Linked Data (Bizer, Heath, and Berners-Lee 2009) is based on the idea of bringing various forms of data to the Web in a standardized and highly interconnected way. In recent years, we have witnessed this originally mostly academic concept increasingly being embraced by large institutions and corporations like the US and UK governments, the British Library, the New York Times, and Best Buy, as well as numerous smaller data publishers. However, as the number of contributors increases, it becomes increasingly important to develop means of quality assurance. This situation is comparable to the regular document-based Web, where it would be quite utopian to assume that all HTML files will adhere to the official standards and remain free of outdated or inaccurate information such as spam.

Linked Data, as the name suggests, is fundamentally based on the idea of interlinking data. The power of Linked Data comes from the fact that information from different sources can flexibly be combined to enable novel applications that individual datasets do not support on their own. Apart from standardized protocols (Bizer, Heath, and Berners-Lee 2009) and data representation formalisms based on RDF and OWL (Manola and Miller 2004), this hinges on the existence of links that connect datasets that would otherwise be isolated, most importantly `sameAs` links that are intended to express that the referents of two identifiers (URIs) are the same in all possible respects. This paper shows that significant numbers of `sameAs` links on the Web do not adhere to the strict official semantics. We present a novel method to analyse large networks of identity links and automatically identify over 500,000 URI pairs that violate identity constraints. We present a thorough theoretical analysis that motivates and justifies our methodology. Finally, we present an algorithm showing that to some extent the problem can be mitigated automatically.

Related Work. Previous studies presented anecdotal observations about the varying use of `owl:sameAs` in Linked Data, as well as theoretical proposals (Halpin et al. 2010; Halpin, Hayes, and Thompson 2011). Empirically, however, they merely assessed 250 `sameAs` links in an Amazon Mechanical Turk experiment. Another line of research (Ding et al. 2010b; 2010a) studied larger graphs of identity links, but only determined general network properties such as degree distributions and URI counts, without analysing their quality. Cudré-Mauroux et al. (2009) presented a probabilistic framework to assess the trustworthiness of publishers of `sameAs` links. This paper presents the first Web-scale quality assessment of `sameAs` links in the wild, using a novel constraint-based method.

There is a large body of work on using various forms of similarity metrics to predict *entirely new* identity links (Euzenat et al. 2011). The constraint method presented here could be incorporated into such systems, enabling them to avoid mistakes in certain cases.

2 Criteria for Identity

Definition. Identity is often described as the relation that only holds between a thing and itself. The semantics of the `owl:sameAs` predicate correspond to the classic definition of identity, which requires

1. Reflexivity ($x = x$)
2. Indiscernibility of identicals: $x = y \longrightarrow (p(x) \longrightarrow p(y))$ for any property p

The indiscernibility of identicals is sometimes viewed as a formalization of Leibniz' classical statement about identity[1].

[1] *Eadem sunt quorum unum in alterius locum substitui potest, salva veritate* (those things are identical of which one can substitute one in place of the other while preserving the truth). The term

Given that p can also refer to the property of being equal to some y, these two criteria together also imply symmetry and transitivity, making identity an equivalence relation.

Still, these criteria do not make identity judgments trivial.

a) It is rarely unambiguously clear what entities one is referring to because of the general problems of identification and reference. One example is the stability of identity over time, especially if all parts have gradually been replaced, as in the examples of the Ship of Theseus, or of the human body, which undergoes a significant level of cell renewal.
b) It may not be clear what universe of properties to quantify over when assessing whether *all properties* are shared.

Linked Data. Hence, even experts sometimes have a hard time agreeing on whether two identifiers (URIs) denote the same entity. For instance, many `sameAs` links have been published between abstract concepts as defined by the SKOS standard and real-world entities as defined in DBpedia (Auer et al. 2007). Some might even agree to a `sameAs` link connecting a beer brewery with the class of all beer bottles produced by that brewery. Certainly not every application requires that `sameAs` links be strict in the Leibnizian sense. However, transferring properties across equivalent URIs can only safely be done if such a strict form of identity is guaranteed. Otherwise, the application may infer that your glass of Jack Daniel's whiskey is a person who was born in 1846 and currently has a net income of 120 million US dollars. To make matters worse, many published `sameAs` links stem from automatic tools that often make serious disambiguation errors, e.g. confusing the US State of Georgia and the sovereign state of Georgia in the Caucasus region.

3 Criteria for Near-Identity and Similarity

Some have proposed weakening the strict requirements of genuine identity, instead only requiring that many but not all properties be shared, leading to a form of near-identity or strong similarity. In this section, we show that conflating genuine identity and strong similarity is not a good idea.

Theorem 1. *Any two distinct physical entities are (i) similar in infinitely many respects, and (ii) dissimilar in infinitely many respects.*

Proof sketch. Let k be the sum of the mass of the two entities (assuming their mass is 0 if they are not physical objects). Following Goodman (1972), they thus both have a mass less than $k+1\text{kg}$, $k+2\text{kg}$, $k+3\text{kg}$, and so on. Similarly, e.g. a given car and a given feather are both probably more than 100 million km away from the sun, both are not liquids, etc. Likewise, given a quality for which the two entities have distinct values k_1, k_2, one can easily also infer an infinite number of differences, e.g. unlike the car, the feather weighs less than 1kg, less than 1.1kg, less than 1.11kg, etc. □

Hence near-identity and similarity only makes sense with respect to certain salient properties. However, there is very strong evidence for the following.

Leibniz' Law, however, is sometimes also used to refer to the opposite direction, the identity of indiscernibles, or to the conjunction of both directions.

Claim 2. *There is no universal agreed-upon way of determining which properties should count as salient in determining near-identity and similarity.*

Numerous counterexamples show that the salience may depend on a number of factors, including the following.

1. Context: Barsalou (1983) found that seemingly dissimilar categories (e.g. children and jewellery) can be judged as highly similar if contextualized with respect to the property of being *things to retrieve from a burning house*.
2. Human assessors: The salience of properties has been found to depend on factors like the age (Shepp and Swartz 1976) and expertise of the assessors. Experts may e.g. use domain-specific knowledge rather than generic visual similarities (Suzuki, Ohnishi, and Shigemasu 1992).
3. Assessment method: Similarity assessments are inconsistent across different means (similarity vs. difference, rating scales vs. binary judgements, etc.) of eliciting them from human subjects (Tversky and Gati 1978).

Moreover, asymmetric similarity judgements have been observed as well (Tversky 1977). For these reasons, it remains important to distinguish genuine identity from less well-defined notions like near-identity and similarity.

4 Method

Unlike near-identity and similarity, genuine identity as described earlier is a symmetric and transitive relationship, which allows us to find inconsistencies using graph-theoretic notions. As will be explained later in greater detail, Fig˙ 1 hints at how the transitively implied identity of two different DBpedia entities may reveal the existence of potentially incorrect `sameAs` links somewhere in the graph. In our study, we aim at performing large-scale analyses of identity links on the Web of Data using constraints. These constraints surely cannot detect all forms of inaccurate links, a task generally considered AI-hard. However, as we will see later, they often are able to locate quite significant inconsistency problems in the data.

Such an analysis is particularly useful because constraint violations can automatically be detected and avoided, not just in the context of this study but also in applications that consume Linked Data. The techniques can likewise be incorporated into automatic linking and data integration tools as additional heuristics of when two clusters should *not* be linked. Our method is based on the following definition.

Definition 3. *Given an undirected graph $G = (V, E)$ with nodes representing entities and edges representing identity, a* ***distinctness constraint*** *is a collection $D_i = (D_{i,1}, \ldots, D_{i,l_i})$ of pairwise disjoint (i.e. $D_{i,j} \cap D_{i,k} = \emptyset$ for $j \neq k$) subsets $D_{i,j} \subset V$ that expresses that any two nodes $u \in D_{i,j}$, $v \in D_{i,k}$ from different subsets ($j \neq k$) are asserted to correspond to distinct entities.*

An index i is used because, in general, there will often be more than one relevant distinctness constraint.

Unique Name Constraints

We formulate such constraints by making unique name assumptions within datasets. The classical Unique Name Assumption in a given knowledge representation formalism

postulates that any two ground terms t_1, t_2 with distinct names are non-identical. Many knowledge bases and ontologies have been formalized in a way such that we can safely assume that two distinct identifiers `Alice` and `Bob` cannot refer to the same person.

The Semantic Web is very different from traditional closed scenarios, because multiple parties can publish data about the same entity using different identifiers. The OWL standard thus does not make a Unique Name Assumption, but instead provides the `owl:differentFrom` property. In practice, however, most publishers do not take the trouble of publishing `owl:differentFrom` statements between every pair of entities. Hence, formally we may not have any fool-proof way of knowing that `dbpedia:Berlin` is different from `dbpedia:London`. Strictly speaking, the possibility remains that all or many of the returned identifiers refer to the same entity.

Fortunately, many data publishers do have a policy of avoiding any duplicates within their datasets. In such cases, we can thus assume that their dataset adheres to an internal unique name assumption in the sense that there are no duplicate identifiers *within* it. For each such dataset, we can thus formulate a separate distinctness constraint $D_i = (D_{i,1}, \ldots, D_{i,l_i})$, where each $D_{i,j}$ is a singleton set containing a different URI from that same dataset.

DBpedia Constraints. Based on Wikipedia, DBpedia (Auer et al. 2007) contains entries for a wide range of different domains and thus is regularly considered one of the main focal points of the Linked Data cloud. Wikipedia uses redirect pages to divert readers from non-existing articles to the most relevant existing articles. Services like sameas.org have gone ahead and created `sameAs` links between all redirects and their targets in DBpedia. This arguably makes sense for name variants (e.g. from "Einstein (physicist)" and "A. Einstein" to "Albert Einstein"). Although many redirect titles actually denote subtopics or related topics (e.g. "Einstein's theory", "God does not play dice" for "Albert Einstein") (de Melo and Weikum 2010a), we opt for a *quasi*-unique name constraint to focus on the truly incorrect cases.

We assume that two DBpedia resources with different titles are distinct, *unless* one is a redirect of the other or one of the two resources is not a valid DBpedia URI. Any redirects are placed in the same $D_{i,j}$ as their redirect target and thus neither distinctness nor identity is assumed between them. URIs that use the DBpedia namespace but do not actually exist in DBpedia are not included in any $D_{i,j}$ at all, for reasons we explain in Section 5.

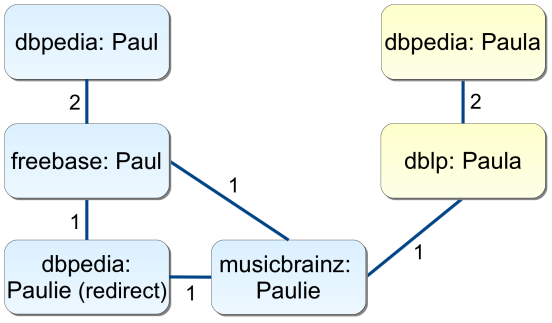

Figure 1: Example of a DBpedia constraint $D_i = (\{$`dbpedia:Paul`, `dbpedia:Paulie(redirect)`$\}, \{$`dbpedia:Paula`$\})$ helping us detect the spurious link from `musicbrainz:Paulie` to `dblp:Paula`

Combinatorial Optimization Problem

If we know that two entities subject to a unique name constraint cannot be identical, we can attempt to remove identity links to disconnect them. In Fig˙ 1, the unique name constraint implies that there are some bad links transitively connecting `dbpedia:Paul` to `dbpedia:Paula`. These two nodes can easily be separated by removing some edges somewhere along the way.

Definition 4. *Given an undirected graph $G = (V, E)$ of identity links and distinctness constraints $D_i 1, \ldots, D_n$ as in Definition 3, a* cut *is a set of edges $C \subset E$ that makes G* consistent *with the D_i if and only if the modified graph $G' = (V, E \setminus C)$ does not contain any path between any two nodes from two different $D_{i,j}$ for any given D_i.*

Definition 5. *Given additional edge weights $w(e)$, an* optimal cut *C is a cut with with minimal $\sum_{e \in C} w(e)$.*

Edge weights $w(e)$ can be defined to quantify the number of `sameAs` links between the two URIs in either direction, or alternatively one could also plug in similarity measures to account for label string similarities and other evidence.

Given the link structure and edge weights, the optimal cut in Fig˙ 1 is the one that deletes the edge between `musicbrainz:Paulie` and `dblp:Paula`. If we only have two nodes s, t to be separated, this corresponds to computing a minimal s-t graph cut C, a problem in P, solvable for instance using the Edmonds-Karp algorithm (Edmonds and Karp 1972).

However, when we have a set $D_{i,j}$ with more than one item, as here for DBpedia, or when a constraint D_i contains more than two sets $D_{i,j}$, or even worse, when we have multiple constraints $D_1, \ldots, D_k$ that all apply simultaneously, each with different sets of nodes $D_{i,j}$, the problem is much more complicated.

Theorem 6. *Computing an optimal $C \subseteq E$ is NP-hard and APX-hard.*

Proof sketch. The minimum multicut problem involves an undirected graph $G = (V, E)$ with edge weights $w(e)$ and a set $\{(s_i, t_i) \mid i = 1 \ldots k\}$ of k demand pairs. The objective is to find a graph cut C that separates each s_i from the respective t_i. Given such a minimum multicut problem, we simply convert each demand pair (s_i, t_i) into a distinctness constraint $D_i = (\{s_i\}, \{t_i\})$.

A gap-preserving problem reduction of this form shows that our problem is at least as hard as the minimum multicut problem, which has been shown to be NP-hard and APX-hard (Chawla et al. 2005). □

Linear Program Relaxation Algorithm

Optimal solutions can be found by transforming the problem into mixed integer linear programs of the following form.

$$\textit{minimize} \sum_{e \in E} d_e w(e) \quad \textit{subject to}$$

$$s_{i,j,v} = 0 \qquad \forall i,\, j{<}l_i,\, v \in D_{i,j} \qquad (1)$$

$$s_{i,j,v} \geq 1 \qquad \forall i,\, j{<}l_i,\, v \in \bigcup_{k>j} D_{i,k} \qquad (2)$$

$$s_{i,j,v} \leq s_{i,j,u} + d_e \qquad \forall i,\, j{<}l_i,\, e \in E,\, u{,}v{\neq}u \in e \qquad (3)$$

$$d_e \in \{0,1\} \qquad \forall e \in E \qquad (4)$$

$$s_{i,j,v} \geq 0 \qquad \forall i,\, j{<}l_i,\, v{\in}V \qquad (5)$$

Decision variables d_e indicate whether $e \in C$, i.e. the identity link represented by e should be removed. Variables $s_{i,j,v}$ indicate the degree of separation of a node v from nodes in $D_{i,j}$. Line (3) ensures that $s_{i,j,v}$ can only be greater/equal 1 if edges along paths are placed in C, and hence line (2) ensures that the solution satisfies all constraints.

Mixed integer linear programming is NP-hard. If we relax (4) to $d_e \in [0,1]$, we obtain a linear program that can be solved in polynomial time. We then use the region growing technique of Garg, Vazirani, and Yannakakis (1996) and de Melo and Weikum (2010b) to efficiently obtain a set of $d_e \in \{0,1\}$ that satisfy logarithmic approximation guarantees with respect to the optimal solution.

Relationship to the Hungarian Algorithm

In the context of aligning two sources using automatic ontology matching algorithms (Euzenat and Shvaiko 2007), the Kuhn-Munkres algorithm (Munkres 1957), also known as the "Hungarian algorithm", has found wide use in improving matching results and ensuring their consistency. Given a weighted bipartite graph, a *matching* or independent edge set is a set of pairwise non-adjacent edges, i.e. a set of edges such that no two edges share a common node. In ontology alignment, a matching thus corresponds to a subset of potential `sameAs` links such that no entity in one dataset is linked to two entities in the other.

The *stable marriage problem* is the problem of finding a *stable matching*, where edges are chosen based on preference rankings, without weights. The Kuhn-Munkres algorithm finds optimal solutions for the *assignment problem* (LSAP, linear sum assignment problem), the task of finding a maximum weight matching, where the total weight of the retained edges is maximized and the total weight of the removed edges is minimized.

Theorem 7. *If $G = (V, E)$ is bipartite with respect to disjoint node subsets V_A, $V_B \subset V$, then computing the minimal cut $C \subset E$ to satisfy two constraints, $D_1 = \{\{u_1\}, \ldots, \{u_{l_1}\}\}$ (for $u_i \in V_A$) and $D_2 = \{\{v_1\}, \ldots, \{v_{l_2}\}\}$ (for $v_i \in V_B$), is equivalent to solving the LSAP.*

Proof sketch. Pairwise non-adjacency implies requiring that connected components never contain more than one node from V_A or more than one node from V_B. This is precisely what the two constraints accomplish. □

Versatility of Algorithm

In comparison to the Hungarian algorithm, our method allows for generalizing the constraints imposed by a bipartite matching between two data sources to an arbitrary number of data sources and additionally explicitly allows for exceptions (e.g. that a URI from one dataset can be linked to two different DBpedia URIs if those two DBpedia URIs are just aliases of each other).

Our algorithm can also incorporate distinctness constraints between specific individual nodes inferred from existing ontological assertions (cf. also Hogan et al. 2012).

1. Explicit distinctness: The OWL `differentFrom` predicate explicitly captures distinctness but is rarely used. Predicates like `disjointWith` and `complementOf` also imply distinctness of the involved classes.
2. Membership in disjoint classes: When two entities have types that are considered disjoint classes, we should be able to infer that they are distinct. Such constraints also help uncover subtle distinctions, e.g. if one entity is the class of all automobiles, another is an individual automobile, and the third is a SKOS conceptual entry.
3. Irreflexive properties (like `flowsInto` for rivers) imply distinctness of the respective s,o in relevant triples (s, p, o), because (s, p, s) and (o, p, o) should not hold.
4. Asymmetric properties (like `properPartOf`) imply distinctness of the respective s, o in relevant triples (s, p, o) because (o, p, s) should not hold.

5 Experiments

Data Preparation and Analysis

Data Sources. We experimented with two real-world data collections:

1. BTC2011: The Billion Triple Challenge 2011[2] Dataset is a large collection of triples crawled from the Web.
2. sameas.org: The sameas.org web site hosts the most well-known collection of coreference links for Linked Data. These have been gathered from many different sources. We used a 2011-05-23 dump of the sameas.org site[3]

Predicates. Table 1 lists relevant properties that we found in the data with unique triple counts (excluding duplicates from different sources). The sameas.org service only publishes `sameAs` links. The BTC 2011 data contains significantly fewer `sameAs` links, which motivates the need for aggregation sites like sameas.org. We see that the more specific forms of identity defined by the OWL standard (`sameIndividualAs`, `sameClassAs`, `samePropertyAs`) are very rare. The properties `equivalentClass` and `equivalentProperty` are more frequent. Their semantics does not require full identity with respect to all properties but only extensional equivalence. Some other related properties that we found include SKOS properties for different degrees of matches and bad URIs like references to a non-existent RDFS `sameAs` property and other misspellings.

[2] We chose the 2011 version to roughly match the version of the sameas.org data that was available to us.

[3] Kindly provided to us by the site's maintainer, Hugh Glaser.

Table 1: Selection of Relevant Properties in BTC 2011

Predicate[1]	Count
BTC2011	
owl:sameAs	3,450,497
w3:2004/02/skos/core#closeMatch	125,313
owl:equivalentClass	25,827
w3:2004/02/skos/core#exactMatch	22,398
owl:disjointWith	3,266
w3:2004/02/skos/core#broadMatch	1,318
w3:2000/01/rdf-schema#sameAs	1,065
owl:complementOf	759
owl:differentFrom	691
owl:equivalentProperty	168
owl:sameIndividualAs	59
biordf:bio2rdf_resource:sameAs	59
w3:2004/02/skos/core#narrowMatch	38
owl:sameClassAs	15
w3:2002/07/owlsameAs	10
owl:samePropertyAs	4
sameas.org	
owl:sameAs	22,411,437

1: where owl: stands for http://www.w3.org/2002/07/owl#, w3: stands for http://www.w3.org/, and biordf: stands for http://bio2rdf.org/.

We could make use of inverse functional properties to infer additional supposed identities. However, in practice, many such properties are not used in a sufficiently clean way. For instance, the `homepage` property of the FOAF vocabulary is defined to be inverse functional, but it has been noted that people often provide the home page of their company, leading to incorrect identifications.

Properties like `disjointWith` could be used to define constraints. Unfortunately, on the Web we also find many incorrect uses of classes, e.g. humans described as being of type OWL ontology, so in many cases the property assertions are wrong rather than the `sameAs` links.

Lastly, we see that `differentFrom` is very rarely ever used, at least too rarely in order to be able to infer that the URIs one usually deals with refer to distinct entities. This is a strong indication that data consumers are implicitly making dataset-specific unique names assumptions.

Graph Construction. For the rest of the study, we focus only on the OWL `sameAs` predicate. We create undirected graphs by determining the symmetric closure of the existing `sameAs` links in the two respective data collections. A third graph was created by combining both data collections. As edge weights, we consider the number of directions in which the sameAs link was encountered (1 or 2).

In the resulting graph for the BTC 2011 `sameAs` triples, the most frequent domains of entity URIs (with respect to the number of edges) belonged to DBpedia, Freebase, lastfm.rdfize.com, linkeddata.uriburner.com, Bibsonomy, and the Max Planck Institute for Informatics, each involved in over 250,000 undirected edges.

Table 2: Constraint Conditions

Dataset	URI Prefix
DBLP	http://dblp.rkbexplorer.com/id/
DBpedia*	http://dbpedia.org/resource/
Freebase	http://rdf.freebase.com/ns/m/
GeoNames	http://sws.geonames.org/
MusicBrainz	http://dbtune.org/musicbrainz/resource/
UniProt	http://bio2rdf.org/uniprot:

$*$: Quasi-unique name constraints with redirect awareness, valid URI checking (DBpedia 3.7) as special conditions

Constraint-Based Analysis

Constraints. We decided to focus on constraint violations with respect to unique names for a small number of major hubs in the Linked Data cloud, listed in Table 2 (additional datasets could easily be added). All URIs matching the given prefix were categorized as belonging to a specific dataset and being subject to its unique name constraint.

DBpedia. Our initial analysis without valid URI checking revealed an enormous amount of constraint violations. In the BTC2011 `sameAs` triples, 205,231 out of 1,055,626 unique DBpedia URIs do not exist in the current DBpedia 3.7 dataset, mainly for the following two reasons.

1. URIs with bad article title escaping: Many datasets contain DBpedia URIs with incorrectly escaped titles, i.e. using a different escaping scheme than DBpedia itself, which results in URIs that do not exist in DBpedia.
2. URIs that no longer exist in DBpedia: As Wikipedia is a living resource, its dynamics may result in changes in article titles as well as deletion of redirects and articles.

This highlights just two examples of the fragility of Linked Data. Given DBpedia's important role in the Linked Data cloud, measures should be taken to address these problems. For instance, an API could be provided to ensure that tools create valid DBpedia URIs and datasets could be released to capture changes to identifiers over time. Since it is not always clear which real DBpedia URI a given invalid URI corresponds to, our DBpedia constraints do not make any claims about such invalid URIs at all.

Constraint Violations. Table 3 presents the detected constraint violations based on the unique name constraints. The total counts of URIs and connected components refer to the preprocessed data consisting only of `sameAs` links and hence do not include singleton URIs that have not been linked to other URIs. Each connected component in the input graph is checked separately for possible constraint violations by intersecting the constraint sets $D_{i,j}$ with the set of nodes in the connected component. If any constraint D_i has more than one non-empty $D_{i,j}$ after intersecting, then the constraint must be violated. In Fig˙ 1, for example, two D_ij of the DBpedia constraint would remain non-empty, indicating that this connected component connects entities from two different D_ij that should be distinct.

Table 3: Constraint Analysis

	BTC2011 +sameas.org	BTC2011	sameas.org
URIs	34,419,740	4,074,166	31,355,505
Connected components	12,735,767	1,387,660	11,853,882
– Average size	2.70	2.94	2.65
Constraint violations			
– node pairs	519,170	138,906	377,057
– node sets	82,309	13,210	71,901
– node sets (DBLP)	25,599	3	25,449
– node sets (DBpedia)	40,691	12,702	31,150
– node sets (Freebase)	407	248	0
– node sets (GeoNames)	15,167	68	0
– node sets (MusicBrainz)	437	181	15,090
– node sets (UniProt)	8	8	212
– affected connected components	81,801	12,974	0

We see that our method is able to detect over half a million node pairs that have been identified although they stem from the same data source and are thus subject to unique name assumptions. The node pairs figures count the number of distinct unordered pairs of nodes that occur within the same connected component, yet are subject to one of the unique name constraints described earlier. In a few instances, constraint violations may stem not from incorrect links but from inadvertent duplicates within a dataset. Fortunately, only in very rare cases would e.g. two duplicate Wikipedia articles and hence DBpedia URIs with different titles exist that describe exactly the same entity in the strict Leibnizian sense.

We additionally list the number of node sets, counting each $D_{i,j}$ for each connected component in which it is actively involved in constraint violations. We include a breakdown by data source, as well as the total number of affected connected components that included constraint violations.

Cleaning. Table 4 presents results regarding the automatic removal of edges to satisfy the constraints used above. To compute the minimal weight graph cuts, we We see that several hundred thousand `sameAs` edges are removed automatically. Note that the number of edges removed is actually lower than the number of constraint violations, because the algorithm explicitly aims at deleting a minimal number of edges in order to ensure that the constraints are no longer violated. When e.g. two densely connected sets of nodes are connected by only a single bad `sameAs` link, detecting and removing that `sameAs` link may satisfy several constraints at once (e.g. between DBpedia entities as well as between GeoNames entities, or e.g. between one DBpedia entity a and several other DBpedia entities b, c, d). In future and related work (Böhm et al. 2012), we are investigating edge weights based on advanced similarity measures to help the algorithm ensure that the correct edges are deleted.

Table 4: Constraint-Based Cleaning

	BTC2011 +sameas.org	BTC2011	sameas.org
Undirected edges removed	280,086	32,753	245,987
Violations per removed edge	1.85	4.24	1.53

6 Implications and Suggestions

Explicit property for strict identity: `sameAs` as it appears in the wild frequently cannot be interpreted as strict identity, and there are no signs of this changing. A separate predicate for genuine identity (e.g. `lvont:strictlySameAs` in the Lexvo.org Ontology (de Melo and Weikum 2008)), while formally declared equivalent to `sameAs`, allows knowing whether a `sameAs` link was indeed intended in the strict sense. or in a looser near-identity sense.

Properties for near-identity/similarity: From a pragmatic perspective, links between entities that are not identical in the strict sense are still important. Despite their apparent vagueness and subjectivity, general notions of near-identity and similarity are useful in many practical applications. Existing examples include SKOS `closeMatch`, the Identity Ontology (Halpin et al. 2010), and the Lexvo.org Ontology (de Melo and Weikum 2008).

Specific relational properties instead of `sameAs`: Due to the subjectivity of near-identity and similarity, we suggest that additional properties be used to describe the exact nature of the relationship holding between different entities when possible. For instance, the relationship between New York City and the New York metropolitan area can be described using a `metropolitanAreaOf` predicate. Studies on polysemy of words have identified fairly common patterns, e.g. the difference between a *university* as an institution vs. a geospatial entity can perhaps be reflected in a `hasPrincipalGeospatialLocation` property that also applies to companies, schools, etc.

7 Conclusion

Clearly, `sameAs` links play an important role in connecting data sets and making the Web of Data more useful. Our study, however, has revealed significant amounts of `sameAs` links that do not adhere to the strict semantics of the OWL standard and hence do not reflect genuine identity. To address this, we have developed a novel method for recognizing and resolving many such cases, based on unique names constraints and a linear program relaxation algorithm.

Additionally, we discussed criteria for identity, near-identity, and similarity from a more theoretical perspective. Moving forward, we propose means of ensuring that both types of use cases – those requiring the strict semantics and those relying on weaker forms – can simultaneously be accommodated in the Linked Data world. Overall, these contributions support applications in benefitting from the Web of Data as it continues to grow.

References

Auer, S.; Bizer, C.; Lehmann, J.; Kobilarov, G.; Cyganiak, R.; and Ives, Z. 2007. DBpedia: a nucleus for a web of open data. In *Proc. ISWC 2007 + ASWC 2007*, Lecture Notes in Computer Science 4825. Springer.

Barsalou, L. W. 1983. Ad hoc categories. *Memory & Cognition* 11(3):211–227.

Bizer, C.; Heath, T.; and Berners-Lee, T. 2009. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.* 5(3):1–22.

Böhm, C.; de Melo, G.; Naumann, F.; and Weikum, G. 2012. LINDA: Distributed web-of-data-scale entity matching. In *Proceedings of the 21st ACM Conference on Information and Knowledge Management (CIKM 2012)*. New York, NY, USA: ACM.

Chawla, S.; Krauthgamer, R.; Kumar, R.; Rabani, Y.; and Sivakumar, D. 2005. On the hardness of approximating multicut and sparsest-cut. In *Proc. 20th IEEE Conference on Computational Complexity (CCC)*, 144–153.

Cudré-Mauroux, P.; Haghani, P.; Jost, M.; Aberer, K.; and De Meer, H. 2009. idMesh: graph-based disambiguation of Linked Data. In *Proceedings of the 18th international conference on World wide web*, WWW '09, 591–600. New York, NY, USA: ACM.

de Melo, G., and Weikum, G. 2008. Language as a foundation of the Semantic Web. In *Proc. ISWC 2008*, volume 401 of *CEUR WS*. Karlsruhe, Germany: CEUR.

de Melo, G., and Weikum, G. 2010a. MENTA: Inducing multilingual taxonomies from Wikipedia. In *Proc. CIKM 2010*, 1099–1108. New York, NY, USA: ACM.

de Melo, G., and Weikum, G. 2010b. Untangling the cross-lingual link structure of Wikipedia. In *Proc. ACL 2010*.

Ding, L.; Shinavier, J.; Finin, T.; and McGuinness, D. L. 2010a. owl:sameAs and Linked Data: An empirical study. In *Proc. 2nd Web Science Conference*.

Ding, L.; Shinavier, J.; Shangguan, Z.; and McGuinness, D. L. 2010b. SameAs networks and beyond: analyzing deployment status and implications of owl:sameAs in Linked Data. In *Proc. ISWC 2010*, 145–160. Berlin, Heidelberg: Springer-Verlag.

Edmonds, J., and Karp, R. M. 1972. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM* 19(2):248–264.

Euzenat, J., and Shvaiko, P. 2007. *Ontology matching*. Springer-Verlag.

Euzenat, J.; Ferrara, A.; van Hage, W. R.; Hollink, L.; Meilicke, C.; Nikolov, A.; Ritze, D.; Scharffe, F.; Shvaiko, P.; Stuckenschmidt, H.; Sváb-Zamazal, O.; and dos Santos, C. T. 2011. Final results of the Ontology Alignment Evaluation Initiative 2011. In *OM*, volume 814 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Garg, N.; Vazirani, V. V.; and Yannakakis, M. 1996. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing (SICOMP)* 25:698–707.

Goodman, N. 1972. Seven strictures on similarity. *Problems and projects* 437–447.

Halpin, H.; Hayes, P. J.; McCusker, J. P.; McGuinness, D. L.; and Thompson, H. S. 2010. When owl: sameAs isn't the same: An analysis of identity in Linked Data. In *International Semantic Web Conference (1)*, 305–320.

Halpin, H.; Hayes, P. J.; and Thompson, H. S. 2011. When owl:sameAs isn't the same redux: A preliminary theory of identity and inference on the Semantic Web. In *LDH*, 25–30.

Hogan, A.; Zimmermann, A.; Umbrich, J.; Polleres, A.; and Decker, S. 2012. Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora. *Web Semant.* 10:76–110.

Manola, F., and Miller, E., eds. 2004. *RDF Primer*. W3C Recommendation. World Wide Web Consortium.

Munkres, J. 1957. Algorithms for the assignment and transportation problems. *J. SIAM* 5:32–38.

Shepp, B. E., and Swartz, K. B. 1976. Selective attention and the processing of integral and non-integral dimensions: A developmental study. *Journal of Experimental Child Psychology* 22(1):73–85.

Suzuki, H.; Ohnishi, H.; and Shigemasu, K. 1992. Goal-directed processes in similarity judgement. In *Proc. 14th Annual Conference of the Cognitive Science Society*, 343–348.

Tversky, A., and Gati, I. 1978. *Cognition and Categorization*. New York: Wiley. chapter Studies of similarity, 79–98.

Tversky, A. 1977. Features of similarity. *Psychological Review* 84:327–352.