Some entities are more equal than others: statistical methods to consolidate Linked Data[⋆]

Aidan Hogan, Axel Polleres, Jürgen Umbrich, and Antoine Zimmermann
Digital Enterprise Research Institute, National University of Ireland, Galway
{firstname.lastname}@deri.org

Abstract. We propose a method for consolidating entities in RDF data on the Web. Our approach is based on a statistical analysis of the use of predicates and their associated values to identify "quasi"-key properties. Compared to a purely symbolic based approach, we obtain promising results, retrieving more identical entities with a high precision. We also argue that our technique scales well—possibly to the size of the current Web of Data—as opposed to more expensive existing approaches.

1 Introduction

In a distributed and collaborative environment like the current World Wide Web, there can be a lot of redundancy across data sources. While redundancy increases noisy or unnecessary information, it can also be an advantage, in the sense that two descriptions of the same thing can mutually complete and complement each other. However, identifying the *same thing* is not a straight-forward task at all, since different identifiers are used for equal entities scattered across different datasets on the current Web of Data.

In the Semantic Web, identical entities can be made explicit by asserting a `owl:sameAs` (resp. `owl:equivalentClass`, `owl:equivalentProperty`) relations between instances (classes, properties, resp.). Entity consolidation on the Semantic Web – as we view it here – thus boils down to the task of identifying `owl:sameAs` relations between instances which are not explicitly related. In the literature, this task is also sometimes referred to as "record linkage" [1], "duplicate identification" [2], "object consolidation" [3], "instance matching" [4], "link discovery" [5, 6], or "Co-reference resolution" [7].

In this paper, we define techniques for entity consolidation that take advantage of statistical information about the way predicates are used throughout the Web of Data in order to assess whether these predicates, along with the values associated with them, are good candidate for identifying instances or, conversely, discriminating them. For the moment, our technique relies on data described in terms of overlapping vocabularies, i.e. sharing common identical properties; as we focus on scale, we have decided for a deliberately simple approach.

The idea that we try to automatically simulate with our approach is based on the following intuitions:

1. we can conclude that two instances are representing the same real-world entity if they share several common property-value pairs;

[⋆] This work is partly funded by Science Foundation Ireland (SFI) project Lion-2 (SFI/08/CE/I1380) and an IRCSET postgraduate scholarship.

2. certain properties are more or less appropriate to disambiguate/consolitate entities;
3. likewise, certain values of some properties are more or less appropriate to disambiguate/consolidate entities.

Particularly, we assume that the necessary information to exploit (Item 2) and (Item 3) can be gathered from statistics about Web data.

As an example, let us consider the case of identifying persons. Two descriptions of unknown persons are representing the same human individual if they describe common properties, such as, eye colour, height, gender, name (Item 1). The gender of a person is usually of limited utility to identify someone, as opposed to the name or address (Item 2). However, if the name is a very common one, such as "Sam Smith" in an English-speaking country, the name property is not enough to identify the person with reasonable certainty (Item 3) and e.g. another property that is non-dissciminating by itself necessarily, such as the gender for instance, may again become discriminating.

Interestingly, these three intuitions can be formalised into an algorithm that we describe in Section 2. Then, we detail its implementation and some improvement to make the process scalable and more efficient in Section 3. In Section 4, we describe a preliminary but promising evaluation of our approach, along with some more general discussion on the feasibility of properly evaluating such a system. Section 5 presents related work and compare it to our approach. In Section 6, we conclude wih important issues still to be solved, possible improvements and future work.

2 Statistical entity consolidation

In this section, we formulate an abstract algorithm for computing a similarity measure on pairs of RDF terms.

Let us first describe some formal notions used throughout this paper. We denote RDF terms by $\mathbf{U}$, $\mathbf{B}$ and $\mathbf{L}$, i.e. the sets of all URIs, blank nodes, and literals, respectively. RDF documents are sets of triples $\langle s\ p\ o\ .\rangle \in \mathbf{B} \cup \mathbf{U} \times \mathbf{U} \times \mathbf{U} \cup \mathbf{B} \cup \mathbf{L}$. For a given RDF document G, we denote by $\mathrm{sub}(G)$ (resp. $\mathrm{pred}(G)$, $\mathrm{obj}(G)$) the set of subjects (resp. predicates, objects) appearing in G. We write RDF documents in the common [1] notation.

Example 1. For illustration purposes, let's consider three documents crawled from the Web containing candidate identifiers for consolidation—*viz.*, `ex1:SamSmith`, `ex2:sam_smith` and `ex3:Sam-Smith`—as follows:

```
ex:SomeDoc dc:creator ex1:SamSmith .
ex1:SamSmith a foaf:Person ; foaf:name "Sam Smith" ;
             foaf:gender "male" ; foaf:homepage ex:JSHompage .
```

[1] Turtle.http://www.w3.org/TeamSubmission/turtle/

```
ex:SomeDoc dc:creator ex2:sam_smith .
ex2:sam_smith foaf:name "Dr. Sam J. Smith" ;
              foaf:homepage ex:JSHompage .
```

```
ex:SomeOtherDoc dc:creator ex3:Sam-Smith .
ex3:Sam-Smith a foaf:Person ; foaf:name "Sam Smith" ;
              foaf:gender "female" .
```

A human able to interpret the above notation will quickly discern that `ex1:SamSmith` and `ex2:sam_smith` *likely* refer to the same person, and that `ex3:Sam-Smith` is a separate person. In this case, a human will intuitively understand that a single document is unlikely to have two authors with the same (first and last) name, and that two people will rarely share a homepage. Sharing the same name and the same homepage are both good indicators that the first and second entities are referring to the same person. However, class membership such as `foaf:Person` does not particularly indicates uniqueness. A human will also understand that the third Sam Smith is female, and that a person usually only has one unique value for gender—thus, the third entity is distinct from the earlier two.

In the following, we try to formalise the above described intuitions such that we can implement an algorithm for performing entity consolidation similar to our fictitious human consumer from this example. A human naturally has the required experience of the world to draw the above conclusions, whereas a machine does not; thus, we must first derive a means of identifying properties and property value pairs which somehow discriminate an entity: *e.g.*, that different entities rarely have the same value for the `foaf:homepage` property. We must then provide a means of translating our knowledge of properties and values into probabilistic equivalence assertions for entities with shared property-value pairs: *e.g.*, that if two entities share a homepage, then there is a probability of p that they are the same. We must further provide a means of aggregating all p values for the same entity pairs to derive an overall score for an equivalence relation between those two entities. Finally, following similar trains of thought we should be able perform disambiguation of entities—again using our statistical knowledge of the usage of properties—to derive a score indicating the likelihood that two entities are *not* equivalent: *e.g.*, that if two candidates initially deemed likely to be equivalent have a different value for `foaf:gender`, then they are likely not equivalent after all. However, in the present paper we only focus on the consolidation part, leaving disambiguation as future work.

2.1 Web Crawl Dataset

To illustrate the type of results produced in our approach, we use a 20M triple RDF Web crawl for which we offer statistics and later derive some evaluation. This dataset was crawled in late January 2010. We also derive some real examples from the dataset in this section. We refer to this dataset as G_{20M}.

2.2 Property-centric statistics

In this paper, we tackle consolidation by relying purely on the statistical characteristics of properties as observed for a given RDF graph. So, to begin, we formalise some statistical characteristics of properties and property-value pairs which approximately quantify how discriminating these are, i.e., to what degree they "identify" the entity to which they are attached.

Thus, when in what follows we speak of *cardinality* for example, it is important to note that we rather intend the notion of an "observed" cardinality—observed with respect to a given graph—in contrast to, *e.g.*, the cardinality explicitly declared within OWL constructs `owl:cardinality`, `owl:minCardinality`, `owl:maxCardinality`, `owl:FunctionalProperty`, `owl:InverseFunctionalProperty`, etc. With this in mind, we now give some preliminary definitions.

Definition 1 (Cardinality). *Let G be an RDF document, p be a property used as a predicate in G and s be a subject in G. The* observed cardinality *(or simply* cardinality*) of p* wrt *s in G, denoted $\mathrm{Card}_G(p, s)$, is the cardinality of the set $\{o \in \mathrm{obj}(G) \mid \langle s\ p\ o\ .\rangle \in G\}$.*

Example 2. Take the graph G_{EX} of all triples from Example 1; the cardinality of the property `dc:creator` with respect to the subject `ex:SomeDoc` is given as $\mathrm{Card}_{G_{EX}}(\texttt{dc:creator}, \texttt{ex:SomeDoc}) = 2$.

We see the cardinality as an initial indicator of how suitable a given pair $< p, s >$ is for discriminating an entity identified by the object. Given a set of cardinalities for a given property, we can define the straightforward notion of *average cardinality* for p as the average of all cardinalities observed for p; *viz*:

Definition 2 (Average cardinality). *Let G be an RDF document, and p be a property used as a predicate in G. The* average cardinality of p, *written $\mathrm{AC}_G(p)$, is the average of the non-zero cardinalities of p wrt a variable s. Formally, $\mathrm{AC}_G(p) = \frac{\sum_{s \in \mathrm{sub}(G)} \mathrm{Card}_G(s,p)}{|\{s \in \mathrm{sub}(G) | \langle s\ p\ o\ .\rangle \in G\}|}$.*

Example 3. Again given G_{EX}, the average cardinality of the property `dc:creator` is given as $\mathrm{AC}_{G_{EX}}(\texttt{dc:creator}) = 1.5$.

Given a property appearing as a predicate in the graph, the corresponding average cardinality is necessarily a positive value greater than one. We may view the average cardinality as roughly corresponding to the probability that two entities identified by a given object are equivalent if they share a given predicate-subject pair—more succinctly, we could interpret properties with average cardinalities close to one as *quasi-functional*.

Given that RDF graphs preserve direction, we can likewise introduce the dual notion of *inverse cardinality* and *average inverse cardinality*, which intuitively coincide with the above definitions replacing subject with object; *viz.*, :

Definition 3 (Inverse cardinality). *Let G be an RDF document, p a predicate in G and o an object in G. The* inverse cardinality of p wrt o in G *is the cardinality of the set $\{s \in \mathrm{sub}(G) \mid \langle s\ p\ o\ .\rangle \in G\}$. This is written $\mathrm{ICard}_G(p, o)$.*

Example 4. Again given G_{EX}, the inverse-cardinality of property `foaf:name` with respect to object `"Sam Smith"` is: $\mathrm{ICard}_{G_{EX}}(\texttt{foaf:name}, \texttt{"Sam Smith"}) = 2$.

Definition 4 (Average inverse cardinality). *Let G be an RDF document, p a predicate in G. The* average inverse cardinality of *p is the average of the non-zero inverse cardinalities of p wrt a variable o. This is written $\mathrm{AIC}_G(p)$. Formally,* $\mathrm{AIC}_G(p) = \frac{\sum_{o \in \mathrm{sub}(G)} \mathrm{Card}_G(p,o)}{|\{o \in \mathrm{sub}(G) | \langle s\ p\ o\ .\rangle \in G\}|}$.

Example 5. Again given G_{EX}, the average inverse cardinality of the property `foaf:name` is: $\mathrm{AIC}_{G_{EX}}(\texttt{foaf:name}) = 1.5$.

In analogy to the above said, we may view the inverse cardinality as an initial indicator of how suitable a given $< p, o >$ pair is for discriminating an entity identified by the subject, and see low average inverse cardinality scores as an indicator for *quasi-inverse-functional* properties.

Please note that hereafter, whenever there is no ambiguity, we conveniently omit the name of the graph in index, writing, *e.g.*, $\mathrm{Card}(p, s)$ instead of $\mathrm{Card}_G(p, s)$.

The above indicators are indeed naïve in terms of quantifying the inverse-functional/functional nature of a given property, and require further tailoring.

Strangely, the *absolute* accuracy of the above metrics are contingent on the consistency of naming for entities—the lack of which is the precise motivation for the metrics; *e.g.*, if we see that seven distinct subjects—which in actuality refer to the same book—have a given object-value for the property `ex:isbn`, we would unduely punish `ex:isbn` by deriving a higher score for the average cardinality. However, we would hope that the more important *relative* accuracy of our metrics in a large enough dataset are not so affected—as long as the metrics for our properties are *proportionately* affected by inconsistent naming, we are not so concerned.

In order to remove obvious noise, we must firstly consider the prevalence of blank-nodes in Linked Data and their effect on our metrics: obviously, by their very nature blank-nodes cannot have any naming consistency across Web documents. For example, the social blogging platform hosted on the `livejournal.com` domain exports large volumes of FOAF[2] data describing users, and only infrequently uses URIs to identify entities; users are given unique blank-node identifiers in each document they appear in. Now, *e.g.*, when the same `foaf:weblog` object-value is given for the same user in several different documents, the average inverse cardinality of `foaf:weblog` is severely and disproportionately increased. In order to improve our initial naïve metrics, we can begin them by simply ignoring blank-node objects when computing average cardinalities and, conversely, ignoring blank-node subjects when computing inverse average cardinalities. We denote these adapted metics excluding blank nodes by Card-XB and AIC-XB, respectively.

Along these lines, in Table 1 we present the average inverse cardinality for the top five of those properties in our Web crawl which are explicitly declared to be inverse-functional (i.e. of type `owl:InverseFunctionalProperty`). Following the

[2] `http://foaf-project.org`

above discussion, we would reasonably expect values close to one; we also show the corresponding values when blank-nodes are ignored as above. Somewhat confirming our suspicion, we can observe that, *e.g.*, the AIC for `foaf:weblog` becomes more accurate when blank-nodes are ignored. We also note that `foaf:mbox` still has a high AIC-XB value due to one source which exports the same `foaf:mbox` values for numerous diverse URI subjects.[3]

IFP	**Occurrences**	**AIC**	**AIC-XB**
`foaf:weblog`	113,091	1.978	1.007
`foaf:mbox_sha1sum`	74,525	1.039	1.014
`foaf:homepage`	72,941	1.016	1.004
`contact:mailbox`	1,272	6.144	1
`foaf:mbox`	1,113	2.338	2.006

Table 1. Average inverse-cardinalities for the top five instantiated properties asserted to be inverse-functional.

We provide similar analysis in Table 2, giving average cardinalities for declared `owl:FunctionalProperty`ies. Again we note that the values approximate one, but we observe that the results are generally less affected by blank-nodes.

FP	**Occurrences**	**AC**	**AC-XB**
`foaf:primaryTopic`	69,072	1.066	1.065
`loc:address`	2,540	1	1
`loc:name`	2,540	1	1
`loc:phone`	2,540	1	1
`foaf:gender`	1,513	1.001	1.001

Table 2. Average cardinalities for the top five instantiated properties asserted to be functional.

Another problem which requires consideration in our metrics is that of incomplete knowledge: given the fact that less observations derive a lower AC/AIC score for a property, we should be more conservative in using less observed properties for consolidation. Thus, we introduce the notion of an *adjusted average cardinality*, where we use a standard credibility formula to dampen averages derived from relatively few observations towards a more conservative mean value [8].

Definition 5 (Adjusted Average Cardinality). *Let p be a property appearing as a predicate in the graph. The* adjusted average cardinality of p *is then* $\mathrm{AAC}(p) = \frac{\mathrm{AC}(p) \times n_{\overleftarrow{p}} + \overline{\mathrm{AC}} \times \overleftarrow{n}}{n_{\overleftarrow{p}} + \overleftarrow{n}}$ *where $n_{\overleftarrow{p}}$ is the number of distinct subjects that appear in a triple with p as a predicate, $\overline{\mathrm{AC}}$ is the average cardinality for all predicate-subject pairs, and $\overleftarrow{n}$ is the average number of distinct subjects for all predicates in the graph.*

Note that above, it may be more intuitive to think of $n_{\overleftarrow{p}}$ as corresponding to the number of observed cardinalities used to derive $\mathrm{AC}(p)$. The above credibility

[3] `http://rdfweb.org/2003/02/28/cwm-crawler-output.rdf`

formula ensures that for AC values derived from a low number of observations ($n_{\overleftarrow{p}} \ll \overleftarrow{n}$), the adjusted AC value is more influenced by the mean $\overline{\text{AC}}$ value than the observed value AC(p); conversely, when $n_{\overleftarrow{p}} \gg \overleftarrow{n}$, the observed AC($p$) value has more influence. From our dataset, for the AAC we observed a value for $\overleftarrow{n}$ of 3985, and a value for $\overline{\text{AC}}$ of 1.153.

We define Adjusted AIC analogously, where $\overline{\text{AIC}}$ denotes the average cardinality for all predicate-object pairs and $\overrightarrow{n}$ is the average number of distinct objects for all predicates in the graph. From our dataset, for the AAIC we observed a value for $\overrightarrow{n}$ of 754, and a value for $\overline{\text{AIC}}$ of 6.094.

Example 6. From G_{20M}, for property `rel:childOf`, AIC(`rel:childOf`)=1.414 and $n_{\overrightarrow{\texttt{rel:childOf}}}$=74. Then, AAIC(`rel:childOf`)=$\frac{74\times1.414+6.094\times754}{74+754}$=5.85: a conservative score reflecting the lack of observations for `rel:childof`.

Taking property `foaf:name`, AIC(`foaf:name`)=1.161 and $n_{\overrightarrow{\texttt{foaf:name}}}$=66,244. Then, AAIC(`foaf:name`)=$\frac{66{,}244\times1.161+6.094\times754}{66{,}244+754}$=1.293: a more confident score reflecting the wealth of observations for `foaf:name`.

2.3 Computing confidence for entity equivalences

We now want to use the cardinalities, inverse cardinalities, AAC and AAIC values of properties and values that are shared by two entities to derive some score indicating the likelihood that those two entities are equivalent; referring back to our running example, the instances `ex1:SamSmith` and `ex2:sam_smith` share the object-value `ex:JSHomepage` for property `foaf:homepage` and the subject-value `ex:SomeDoc` for the property `dc:creator`—similarly, `ex1:SamSmith` and `ex3:Sam-Smith` share the object-value `"Sam Smith"` for property `foaf:name`. To do this, we need a metric which combines the (inverse) cardinality and the AA(I)C score for a given property-value pair, where the former value indicates the "uniqueness" of the value for the property, and the latter value gives a more general indication of the (inverse-) functional nature of the property.

We start by assigning a coefficient to each pair $\langle p, o\rangle$ and each pair $\langle p, s\rangle$ that occur in the dataset, where the coefficient is an indicator of how much the pair helps determining the identity of an entity. In particular, for the purposes of later aggregation, we require the coefficient to be a positive value less than one. We determine the coefficient for a $\langle p, s\rangle$ pair as $C(p,s) = \frac{1}{\text{Card}(p,s)\times\text{AAC}(p)}$, and the the coefficient for $\langle p, o\rangle$ as $C^{-}(p,o) = \frac{1}{\text{ICard}(p,o)\times\text{AAIC}(p)}$.

Example 7. Take $G_{EX'}$ as a version of G_{20M} which contains G_{EX}—essentially, we want to refer to the running example using real statistics from our evaluation. Take AAIC$_{G_{EX'}}$(`foaf:name`)=1.293 as before.

Now, let us speculate that ICard$_{G_{EX'}}$(`foaf:name`, `"Sam Smith"`) = 7, reflecting in this example that "Sam Smith" is somehow a relatively common name. Then, C^{-}(`foaf:name`, `"Sam Smith"`) = $\frac{1}{7\times1.293}$ = 0.11.

Now, speculate that ICard$_{G_{EX'}}$(`foaf:name`, `"Dr. Sam J. Smith"`) = 2, reflecting in this example that the name "Dr. Sam J. Smith" is more rare. Then, C^{-}(`foaf:name`, `"Dr. Sam J. Smith"`) = $\frac{1}{2\times1.293}$ = 0.387 .

With coefficients for each property-value pair at hand, we can now derive an aggregated confidence score for entity equivalences. To this end, we define the following aggregation function:

Definition 6 (Aggregated Confidence Score). *Let* $Z = (z_1, \dots z_n)$ *be a non-empty n-tuple such that* $Z \in [0,1]^n$ *and let* $\max \in [0,1]$. *The aggregated confidence value* $\mathrm{ACS}(Z, \max)$ *is computed iteratively: starting with* $\mathrm{ACS}^0 = 0$, *then for each* $k = 1 \dots n$, $\mathrm{ACS}^k = (\max - \mathrm{ACS}^{k-1}) z_k + \mathrm{ACS}^{k-1}$.

The above confidence function is commutative (wrt. the order of z_i, z_j) and produces a value between 0 and max inclusive. Taking max as 1, the main idea is to view Z as a list of probabilistic scores for a given observation, and that each successive score ACS^k reduces the uncertainty $1 - \mathrm{ACS}^{k-1}$ by a product of the current observation z_k—we parameterise max for full flexibility of the aggregation function. Also, the function gives higher weight to more certain observations. Indeed, take $Z_a = (0.5, 0.5)$ and $Z_b = (0.9, 0.1)$; $\mathrm{ACS}(Z_a, 1) = (1 - 0.5) \times 0.5 + 0.5 = 0.75$ whereas $\mathrm{ACS}(Z_b, 1) = (1 - 0.1) \times 0.9 + 0.1 = 0.91$.

To compute the aggregated confidence score for the equivalence of two entities e_1, e_2, we first define the sequence of subject equivalence coefficients $s^{e_1,e_2} = (s_1^{e_1,e_2}, \dots, s_n^{e_1,e_2})$ as an ordering of the multiset $\{C^-(p,o) \mid \langle e_1\ p\ o\ .\rangle \in G \wedge \langle e_2\ p\ o\ .\rangle \in G\}$—that is, the coefficients for pairs $\langle p, o\rangle$ that appear in a triple with subject e_1 as well as in a triple with subject e_2. We define the sequence of object equivalence coefficients $o^{e_1,e_2} = (o_1^{e_1,e_2}, \dots, o_n^{e_1,e_2})$ analogously via $C(p,s)$.

Let Z_{e_1,e_2} be the concatenation of the sequences s^{e_1,e_2} and o^{e_1,e_2}, that is, $Z_{e1,e2}$ represents the confidences derived from the coefficients of all property-value pairs shared by the two entities. We could now naïvely compute the aggregated confidence score as $\mathrm{ACS}(Z_{e1,e2}, 1)$.

Example 8. Again take $G_{EX'}$, where $\mathrm{AAIC}_{G_{EX'}}(\texttt{foaf:homepage}) = 1.068$ and $\mathrm{AAC}_{G_{EX'}}(\texttt{dc:creator}) = 1.214$. Further, let us assume $\mathrm{ICard}_{G_{EX'}}(\texttt{foaf:homepage}, \texttt{ex:JSHompage}) = 2$ and $\mathrm{Card}_{G_{EX'}}(\texttt{dc:creator}, \texttt{ex:SomeDoc}) = 2$. As before, we can determine $C^-(\texttt{foaf:homepage}, \texttt{ex:JSHompage}) = 0.468$ and $C(\texttt{dc:creator}, \texttt{ex:SomeDoc}) = 0.412$.

Now, taking `ex1:SamSmith` and `ex2:sam_smith` as candidates for consolidation, we can determine $Z_{\texttt{ex1:SamSmith},\texttt{ex2:sam\_smith}} = (0.468, 0.412)$, and finally compute $\mathrm{ACS}(Z_{\texttt{ex1:SamSmith},\texttt{ex2:sam\_smith}}, 1)$=0.687.

However, the above aggregation is still too naïve for Web data in that it assumes that observations based on property-value pairs are completely independent. As a counter-example, we present Figure 1 which shows a real sample taken from our crawl in which we see two people share some relation to six distinct subject/object values. We observe a clear correlation between these properties.

Firstly, we must consider that two entities which share at least one value for a given property are more likely to share subsequent values; thus, we cannot view the subsequent readings as independent observations, and must take into account possible correlation: *e.g.*, two people who have co-authored at least one paper together are more likely to co-author more. Thus, as a counter measure,

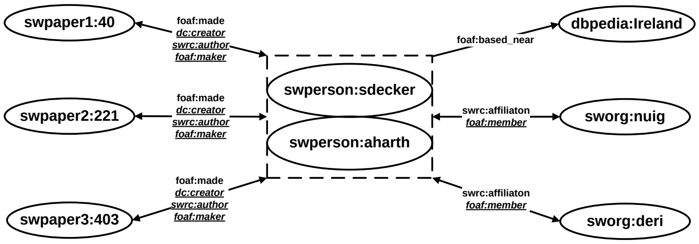


Fig. 1. Real example of inter- and intra-property correlation

for the observed shared property-value pairs for e_1 and e_2, we first aggregate the values for each property p_k (in each direction) separately using the above aggregation function: during this aggregation, we set the max value to $\frac{1}{AC(p_k)}$ or $\frac{1}{AIC(p_k)}$ respectively. Thus, for the example presented in Figure 1, we would only allow, *e.g.*, `dc:creator` to contribute a total value of 0.824. We then perform the aggregation function again over the individual scores of all properties (in each direction).

Aside from correlation for values on a single property, there may also be correlation between different properties—*e.g.*, sub-properties or inverse-properties—which relate two entities to the same external literal or entity. Thus, we prune our observations whereby if we have multiple properties connected to the same term, we keep the property with the lowest AC(p) or AIC(p) value for either direction, and remove consideration of all other properties.

The above two steps to counter-act "obvious" correlation reduced the aggregated confidence scores for the two entities presented in Figure 1 from 0.969 in the naïve case, to 0.781. Admittedly, the new confidence is still quite high—one could further try to detect and account for less obvious forms of correlation such as between a person's affiliation, location and co-authors. However, such considerations are outside of the current more preliminary scope.

3 Implementation

In order to simplify discussion of our implementation, we solely refer to the calculations based on AIC until necessary. Calculations based on AC are directly analogous, where object and subject are simply swapped.

We wish to see our methods used at scale over Linked Data, thus we attempt to use scalable operations to implement our statistical analysis: specifically, we rely mainly on sorts and scans. Data is stored in N-Triples (or possibly N-Quads) format in a flat GZipped compressed file.

Assuming an input unsorted dataset, our first step is to sort the data according to the following lexicographic order (using a merge-sort):

$$(p, o, s)$$

The data is thus grouped according to common p values, and further according to common p, o pairs. Thus, we can calculate the inverse-cardinality for each p, o by means of a scan. Further, by storing the distribution of inverse-cardinalities observed for a given property, we can similarly compute the average inverse cardinality for each property on the fly. Thus, we perform a single scan of the ordered data and extract all of the cardinality information needed for the proceeding steps, as well as the $\overleftarrow{n}$ and $\overline{\mathrm{AC}}$ figures required for the credibility formula.

We can then perform a second scan of the same data, this time using the statistics produced in the first scan to derive initial confidence scores for each individual *po* pair. That is to say, we can use the $AIC(p)$ and $\mathrm{ICard}(p, o)$ to compute $C^{-}(p, o)$ values, and propagate these values as initial indicators of equivalence for subjects with the same $\langle p, o \rangle$. Thus, after the second scan we produce the following tuples:

$$(e_1, e_2, C^{-}(p, o), p, o, -)$$

These tuples are written again to a new compressed file (in general N-Triple form); note that the '$-$' is simply to indicate direction of the observation.

Applying the exact same process over data ordered by: (p, s, o), we can also derive tuples of the form:

$$(e_1, e_2, C(p, s), p, s, +)$$

Note that we do not produce reflexive or symmetric versions of the above tuples: for the above tuples, e_1 will always be less than e_2 with respect to the given lexicographical order which allows us to halve the set of tuples, while ensuring consistency in tuple "naming". Indeed, the production of such tuples is quadratic with respect to the given input which naïvely could seriously hamper scalability we aim for. In order to illustrate this, Figure 2(a) and Figure 2(b) show the cumulative increase in tuples when considering increasing sizes of "equivalence classes" derived for increasingly common p, o and p, s pairs respectively. Conveniently however, the increased equivalence class sizes corresponds to a higher inverse-cardinality/cardinality values, which implies that the common $\langle p, o \rangle / \langle p, s \rangle$ pairs which produce the larger equivalence classes are in any case useless for consolidation in our scenario. For the moment, we implement an arbitrary threshold and throw away equivalence tuples derived from $\langle p, o \rangle / \langle p, s \rangle$ pairs with s/o values greater than 100.

Finally, both incomplete sets of tuples can then be merge-sorted to produce a file grouped by e_1 and e_2. The sorted tuples can then be scanned, with the above aggregation functions being applied for each $\langle e_1, e_2 \rangle$ pair.

We deem the above methods to be relatively scalable—with the caveat of quadratic equivalence tuples being produced—again, with a sensible threshold, such explosion of output can be mitigated. In any case, we admittedly have yet to test our methods with respect to performance or scale on larger datasets with varying thresholds. For the moment, we focus on some quality evaluation to ensure that our approach derives some reasonable results.

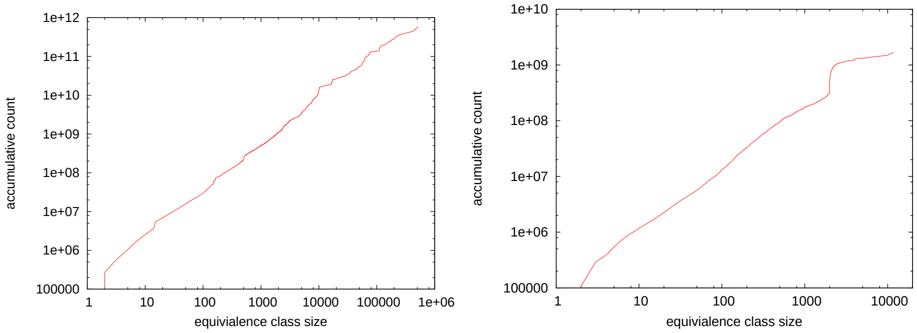


Fig. 2. Cumulative increase in tuples when considering increasing sizes of "equivalence classes" derived for increasingly common p, o and p, s pairs respectively.

4 Quality evaluation

The evaluation of our approach is problematic because there is no existing benchmark for consolidation of Web data. We nonetheless tried two different approaches.

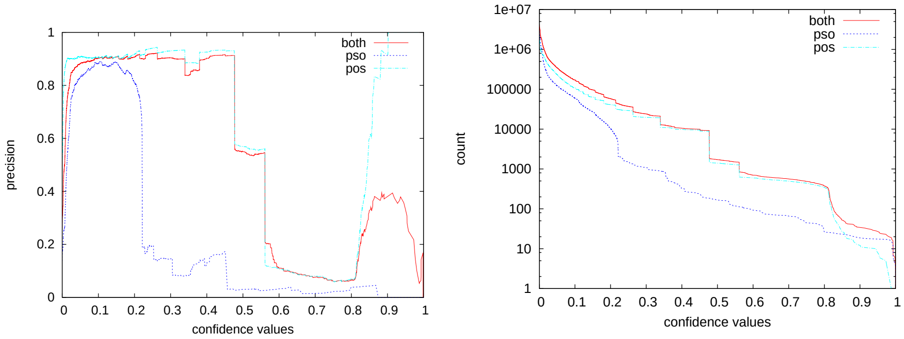
Firstly, we extract our own "best-effort" benchmark from our crawl of 20 million triples by the following process:

- we extract asserted `owl:sameAs` statements and infer additional `owl:sameAs` statements using the same technique as in [3]—a single iteration of reasoning using `owl:FunctionalProperty` and `owl:InverseFunctionalProperty` assertions;
- we separate all `owl:sameAs` statements and additionally compute the transitive closure thereof;
- we prune the dataset by keeping only triples which have either a subject or an object that appears in a `owl:sameAs` statement;
- again, we discard the `owl:sameAs` statements which relate an entity for which we have no information;
- we again finally prune the dataset removing triples for which the subject or object do not have `owl:sameAs` statements.

The resulting set of `owl:sameAs` statements contains 36,134,230 transitively closed, non-symmetric, non-reflexive (reflecting the nature of the same-as output of our statistical approach) `owl:sameAs` statements over 87,586 entities. The evaluation data consists of 5,622,898 triples. We view the derived asserted/inferred `owl:sameAs` statements as a partial ground-truth for our quality evaluation: please note that we are aware of the somewhat ironic nature of our evaluation approach—if we apply our previous work on reasoning, we would achieve a perfect 100% recall and 100% precision. However, again this evaluation is best-effort, and is intended in this preliminary analysis to present illustrative statistics about the precision of our approach in the spirit of a proof-of-concept.

Along these lines, in Figure 3(a) we present the precision of our approach considering AIC values, AC values, and both values combined. Indeed, our precision is quite high at even low levels of confidence, reaching roughly 92% at a confidence value of 0.26. However, our approach suffers from deriving a small number of incorrect equivalences at high confidence. Severe drops in precision are due to the derivation of large numbers of correct inferences at an exact precision; *e.g.*, we derive 630 correct inferences at the precise value of 0.6777389199225334—all uniformly described entities found in the aforementioned `livejournal.com` domain. Thus, once we go above that threshold, the precision severely drops. Essentially, large volume equivalences are derived at lower confidence values, and incorrect equivalences between entities described in smaller exporters are derived at higher confidence values. Figure 3(b) is presented for cross-reference, where the amount of remaining correct inferences drop in correlation with the drops in precision from Figure 3(a). Interestingly, from Figure 3(a) we can conclude that considering AIC values alone approximates consideration of both directions.

With respect to recall, we observed a value of about 3% with respect to the transitively closed ground truth. However, one should note that we do not perform any transitive closure over the output of our statistically derived equivalences, and thus it is difficult to derive an adequate recall comparison.



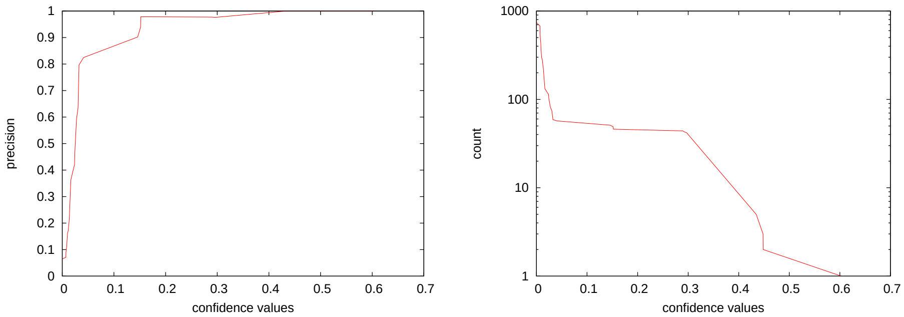
(a) Precision of our approach considering AIC (pos), AC (ps) and both, with respect to different threshold of confidence

(b) Correct equivalences found considering AIC (pos), AC (ps) and both, with respect to different levels of confidence

In our second evaluation, we used our consolidation approach as an instance matching tool by selecting only the consolidation that matches named terms from two distinct datasets. This method has the merit of being comparable to other instance matching algorithms over the reference datasets of the Ontology Alignment Evaluation Initiative[4]. The OAEI offers a well established competition in the ontology matching community, and an instance matching track was added in 2009. The drawback of this method is that the datasets used are very homogeneous (3 sets of bibliographic data) and are using the same terms in a very similar

[4] `OAEI.http://oaei.ontologymatching.org/`

way to each other. Therefore, they are not representative of what is really found on the Web of Data. The results in Figure 4 shows that we get a much lower recall than specialised instance matching tools (cf. [9], Fig. 12, p40). We do not consider that this demonstrate a flaw of our approach. On the contrary, we think that it shows the limits of evaluating a generic consolidation approach with a specific, small-scale instance matching dataset. Unfortunately, a solid evaluation standard for entity consolidation is yet to be devised. Our previous home-made benchmark is an attempt in that direction. In this experiment, AC did not contribute at all to the overall confidence, because of the particular morphology of the data.



(c) Precision in function of the threshold of confidence

(d) Correct equivalences found in function of the threshold of confidence

5 Related work

In our previous work, we used reasoning (functional properties, inverse functional properties, cardinality restrictions) to consolidate Web data with limitation both in terms of precision and recall [3]. Entity consolidation has an older related stream of research relating largely to databases, with work under the names of record linkage, instance fusion, and duplicate identification; cf. [1, 10, 11] and a survey at [2]. Due to the lack of formal specification for determining equivalences, these older approaches are mostly concerned with probabilistic methods. Bouquet et al. [12] motivate the problem of (re)using common identifiers as one of the pillars of the Semantic Web, and provide a framework and fuzzy matching algorithms to fuse identifiers. Online systems such as Sig.Ma[5], rkbexplorer[6], and ObjectCoref offer on-demand querying for `owl:sameAs` relations found for a given input URI, which they internally compute and store. Another related field which gained more recent attention is Instance matching. Some references include matching database instances [13], domain-dependent similarity of instances [14, 15], [4], instance matching and linking guided by a "linking language" (Silk) for

[5] `http://sig.ma`

[6] `http://www.rkbexplorer.com/sameAs/`

Linked Data [5]. Also, in 2009, the Ontology Alignment Evaluation Initiative[7] has introduced a new test track on instance matching[8].

6 Discussion and conclusion

Indeed, our work is quite preliminary, and there are many open questions. Firstly, in order for such an approach to be proven useful, we would need to demonstrate that the statistical approach presented can generate additional equivalence relations than standard reasoning approaches. Such was not possible given the nature of our evaluation setups. In theory however, we believe that the presented approach should be able to conclude additional equivalences, and in future work we would need to devise a means of evaluating such. Similarly, we should also incorporate reasoning approaches into the current statistical model, developing a hybrid approach which hopefully generates more equivalences.

Perhaps a more interesting use-case for statistical approaches is for disambiguating entities: that is, stating that two entities are different. Such `owl:differentFrom` relations are rarely specified on the Web—they can however be inferred from, *e.g.*, more common `owl:disjointWith` assertions. Given a set of candidate equivalences derived through reasoning, statistical or hybrid approaches, disambiguation can be applied to improve precision of results; reasoning on, *e.g.*, `owl:InverseFunctionalProperty` assertions is known to be imprecise [3]—clearly, our approach could also benefit from some disambiguation post-processing. Indeed, one could consider an iterative approach, where the confidence scores for equivalence and difference are iteratively refined—and statistics are iteratively made more accurate—until a satisfactory fixpoint.

Further, we would intend to evaluate the performance characteristics of our approach on larger datasets, with the aim of applying the analysis over a dataset in the order of a billion triples. Again, our approach is based on a scalable substrate of sorts and scans, and so we would see this as a feasible goal.

With respect to improving the algorithms presented herein, we would need to consider more advanced topics. Perhaps the most important is the consideration of the source of data when deriving statistics. The statistics for usage of properties is heavily influenced by large RDF exporters on the Web. Most of the incorrect highly-confident equivalences were the result of applying such statistics over smaller heterogeneous sources. One might argue that there currently is not enough heterogeneous Linked Data to give enough confidence for such statistical approaches—the "reasonable ineffectiveness of Linked Data" if you will; however, we should still attempt to consider some notion of a "dataset" as a grouping of uniform RDF data—*e.g.*, published by the same exporter—and consider a weighted version of our statistics which includes such a concept.

Also, we would have to look at deriving some form of transitive closure over the 'fuzzy' equivalences produced to improve recall. The exact nature of such a closure is the topic for future research. Similarly, detection of some notion of correlation

[7] `OAEI.http://oaei.ontologymatching.org/`

[8] Instance data matching. `http://www.scharffe.fr/events/oaei2009/`

between properties—besides the more obvious cases already discussed—is worthy of further investigation, and would be useful to ensure more sensible aggregation of confidence scores. Other topics, such as fuzzy string matching techniques or string-normalisation pre-processing, would also be worth further analysis.

To summarise, we defined a new approach towards consolidating data in a very heterogeneous environment (the Semantic Web at large). We have barely scratched the surface but can already attest that the results are promising.

References

1. Newcombe, H.B., Kennedy, J.M., Axford, S.J., James, A.P.: Automatic Linkage of Vital Records: Computers can be used to extract "follow-up" statistics of families from files of routine records. Science **130**(3381) (1959) 954–959
2. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate Record Detection: A Survey. IEEE Trans. on Knowl. & Data Eng. **19**(1) (2007) 1–16
3. Hogan, A., Harth, A., Decker, S.: Performing Object Consolidation on the Semantic Web Data Graph. In: Proc. of the WWW2007 Workshop I^3. Volume 249 of CEUR Workshop Proceedings., CEUR (2007)
4. Castano, S., Ferrara, A., Montanelli, S., Lorusso, D.: Instance Matching for Ontology Population. In: Proc. of SEBD 2008. (2008) 121–132
5. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and Maintaining Links on the Web of Data. In: Proc. of ISWC 2009. Volume 5823 of LNCS., Springer (2009) 650–665
6. Hassanzadeh, O., Lim, L., Kementsietsidis, A., Wang, M.: A Declarative Framework for Semantic Link Discovery over Relational Data. In: Proc. of WWW 2009, ACM Press (2009) 1101–1102
7. Glaser, H., Jaffri, A., Millard, I.: Managing Co-reference on the Semantic Web. In: Proc. of LDOW 2009. (2009)
8. Whitney, A.W.: The theory of experience rating. In: Proceedings of the Casualty Actuarial Society. Volume 4. (1918) 274–292
9. Euzenat, J., Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Malaisé, V., Meilicke, C., Nikolov, A., Pane, J., Sabou, M., Scharffe, F., Shvaiko, P., Spiliopoulos, V., Stuckenschmidt, H., Sváb-Zamazal, O., Svátek, V., dos Santos, C.T., Vouros, G.A., Wang, S.: Results of the Ontology Alignment Evaluation Initiative 2009. In: Proc. of OM 2009. Volume 551 of CEUR., CEUR (2009)
10. Michalowski, M., Thakkar, S., Knoblock, C.A.: Exploiting Secondary Sources for Automatic Object Consolidation. In: Proceeding of 2003 KDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation. (2003) 34–36
11. Chen, Z., Kalashnikov, D.V., Mehrotra, S.: Exploiting relationships for object consolidation. In: Proc. of IQIS 2005, ACM Press (2005) 47–58
12. Bouquet, P., Stoermer, H., Mancioppi, M., Giacomuzzi, D.: OkkaM: Towards a Solution to the "Identity Crisis" on the Semantic Web. In: Proc. of SWAP 2006. Volume 201 of CEUR Workshop Proceedings., CEUR (2006)
13. Bernstein, P.A., Melnik, S., Mork, P.: Interactive Schema Translation with Instance-Level Mappings. In: Proc. of VLDB 2005, ACM Press (2005) 1283–1286
14. Albertoni, R., Martino, M.D.: Semantic Similarity of Ontology Instances Tailored on the Application Context. In: Proc. of OTM 2006, Part I. Volume 4275 of LNCS., Springer (2006) 1020–1038
15. Albertoni, R., Martino, M.D.: Asymmetric and Context-Dependent Semantic Similarity among Ontology Instances. Jour. on Data Semantics **4900**(10) (2008) 1–30