Entity Deduplication on ScholarlyData

Ziqi Zhang[1], Andrea Giovanni Nuzzolese[2], and Anna Lisa Gentile[3(✉)]

[1] Nottingham Trent University, Nottingham, UK
ziqi.zhang@ntu.ac.uk
[2] Semantic Technology Lab, ISTC-CNR, Rome, Italy
andrea.nuzzolese@istc.cnr.it
[3] IBM Research Almaden, San Jose, CA, USA
annalisa.gentile@ibm.com

Abstract. ScholarlyData is the new and currently the largest reference linked dataset of the Semantic Web community about papers, people, organisations, and events related to its academic conferences. Originally started from the Semantic Web Dog Food (SWDF), it addressed multiple issues on data representation and maintenance by (i) adopting a novel data model and (ii) establishing an open source workflow to support the addition of new data from the community. Nevertheless, the major issue with the current dataset is the presence of multiple URIs for the same entities, typically in persons and organisations. In this work we: (i) perform entity deduplication on the whole dataset, using supervised classification methods; (ii) devise a protocol to choose the most representative URI for an entity and deprecate duplicated ones, while ensuring backward compatibilities for them; (iii) incorporate the automatic deduplication step in the general workflow to reduce the creation of duplicate URIs when adding new data. Our early experiment focused on the person and organisation URIs and results show significant improvement over state-of-the-art solutions. We managed to consolidate, on the entire dataset, over 100 and 800 pairs of duplicate person and organisation URIs and their associated triples (over 1,800 and 5,000) respectively, hence significantly improving the overall quality and connectivity of the data graph. Integrated into the ScholarlyData data publishing workflow, we believe that this serves a major step towards the creation of clean, high-quality scholarly linked data on the Semantic Web.

1 Introduction

ScholarlyData [16] is the evolution of the Semantic Web Dog Food (SWDF) dataset[1]. So far it has taken care of refactoring data from the SWDF dataset at schema level, migrating data representation from the Semantic Web Conference (SWC) Ontology[2] to the new *conference-ontology*[3]. Moreover a workflow is in

[1] http://data.semanticweb.org.
[2] http://data.semanticweb.org/ns/swc/swc_2009-05-09.html.
[3] http://w3id.org/scholarlydata/ontology/.

E. Blomqvist et al. (Eds.): ESWC 2017, Part I, LNCS 10249, pp. 85–100, 2017.
DOI: 10.1007/978-3-319-58068-5_6

place - cLODg[4] (conference Linked Open Data generator) [4] - to ease the tasks of data acquisition, conversion, integration, augmentation, which has already been used to add novel data to the portal[5].

Nevertheless, the problem of verifying data at instance level has not been tackled so far. Despite existing guidelines for populating the SWDF dataset in order to maximise the reuse of existing URIs and minimise the introduction of redundant URIs, these are, in practice, not carefully followed, resulting in a number of duplicate URIs in the dataset. This is especially common for people with multiple names and surnames, which sometime are reported inconsistently in different papers and in general for organisations, for which different variations of the name are often reported. In the initial refactoring from SWDF to ScholarlyData, while we tackled data model issues, we ignored the problems at instance level and we simply kept URIs as is.

In this work we address these issues and extend the cLODg workflow with data integration and verification steps. The aim is to have a procedure in place that when a new data graph is to be added to ScholarlyData, determines for each new URI in the new data graph, whether it refers to an existing entity in ScholarlyData with a different URI, or a truly unknown and new entity. We propose a three steps approach that for each input URI from the new data graph (i) creates a candidate pool of URIs that could potentially represent the same entity in the ScholarlyData dataset; (ii) discovers truly duplicate URIs (if any) for the input URI using supervised classification methods and (iii) finally resolves the duplicates and merges the new data graph into ScholarlyData.

Ultimately, we aim to ensure that each entity in ScholarlyData is represented by one unique URI, which is currently not the case. Therefore in this work, we also apply the proposed method to the existing ScholarlyData dataset in a one-off cleaning process to resolve and deprecate duplicate URIs. The contributions of this work are threefold. First, we expand the data publication workflow for ScholarlyData, adding automatic data integration capabilities. Second, we propose an efficient deduplication method for the scholar domain and show that it outperforms several state-of-the-art models significantly. Finally, we add a maintenance step in the general publication workflow that empirically determines how to resolve duplicate URIs, integrate information, while ensuring backward compatibility.

The paper is structured as follows. Section 2 examines related work; Sect. 3 describes the proposed method; Sect. 4 evaluates our method on a manually annotated dataset and discusses the deduplication results on the ScholarlyData dataset; and finally Sect. 5 concludes the paper and identifies future work.

2 Related Work

Scholarly data. The first considerable effort to offer comprehensive semantic descriptions of conference events is represented by the metadata projects at

[4] cLODG is an Open Source tool that provides a formalised process for the conference metadata publication workflow https://github.com/anuzzolese/cLODg2.

[5] The tool as been used for generating data for ISWC2016 and EKAW2016.

ESWC 2006 and ISWC 2006 conferences [13], with the Semantic Web Conference Ontology being the vocabulary of choice to represent such data. Increasing number of initiatives are pursuing the publication of data about conferences as Linked Data, mainly promoted by publishers such as Springer[6] or Elsevier[7] amongst many others. For example, the knowledge management of scholarly products is an emerging research area in the Semantic Web field known as Semantic Publishing [19]. Numerous initiatives are aimed at offering advanced exploration services over scholarly data, such as Rexplore [17], Open Citation[8], Google scholar[9] and DBLP[10].

Despite these continuous efforts, it has been argued that lots of information about academic conferences is still missing or spread across several sources in a largely chaotic and non-structured way [1]. Improving data quality remains a major challenge with scholarly data. This can include, amongst others, tasks of dealing with data-entry errors, disparate citation formats, enforcement of standards, ambiguous author names and abbreviations of publication venue titles [10]. The ScholarlyData dataset as well currently suffers from data quality issues, mainly due to duplicates, inconsistencies, misspelling and name variations. This work aims at filling this gap, by proposing a strategy to detect co-referent entities and resolve the duplicate URIs in scholarly datasets, specifically on ScholarlyData.

Entity deduplication. Entity deduplication looks for (nearly) duplicate records in relational data, usually amongst data points that belong to the same entity type. This is related to a wide range of literature, such as named entity coreference resolution in Natural Language Processing [22], and Link Discovery on the Semantic Web [14]. The typical approach depends on measuring a degree of 'similarity' between pairs of objects using certain metrics, then making a decision about whether a pair or groups of pairs are mutually duplicate. This is often done using strategies based on similarity threshold, classification (e.g., [2]) or clustering (e.g., [12]) methods. In [14], 10 most recent state-of-the-art systems are evaluated. Our work is most relevant to SILK [8] and LIMES [15], which include supervised models to overcome the arbitrary decision making of link-matching thresholds. Compared to SILK and LIMES, our method differs in terms of the learning algorithms and similarity metrics. In particular, our method deals with the situation where one URI is used much more frequently than its duplicates, which is found to be common in linked datasets [21] and making conventional set-overlap based measures ineffective. Also, the machine learning algorithms in LIMES (e.g., WOMBAT) only use positive examples for training while both SILK (i.e., the genetic algorithm in [8]) and our method can benefit from negative training examples. The 2013 KDD CUP [20] proposed an academic paper author linking task that drew 8 participating systems.

[6] http://lod.springer.com/wiki/bin/view/Linked+Open+Data/About.
[7] http://data.elsevier.com/documentation/index.html.
[8] http://opencitations.net/.
[9] https://scholar.google.com.
[10] http://dblp.uni-trier.de/.

However, the dataset used are very different and hence does not represent the problems in ScholarlyData. Duplicates are found very frequent, often as a result of parsing errors when importing data from different sources, and the winning system heavily relied on ad hoc pre-processing of author names. As we shall show later, duplicates in ScholarlyData are much less frequent and thus challenging to detect.

If addressed in a brute force fashion, the matching task has quadratic complexity as it requires pairwise comparisons of all the records. A common way to reduce complexity is the use of so-called *blocking strategies* (grouping records in blocks before comparing them) to reduce the search space. An approach is to exploit specific criteria in the data schema (the values of particular attributes) to split the data [7,18], as in the Sorted Neighborhood Method (SNM) [7] which performs sorting of the records according to a specifically chosen blocking key. Content based blocking strategies usually look for common tokens between two entities [18] and group entities that share some token(s). Finally, another solution to speed up the comparison step is to map the original entity representations in a lower dimensional space. The most prominent example is *Locality-Sensitive Hashing (LSH)*, which produces effective entity signatures [3]. In this work, we experiment with SNM [7] and a content based approach. We show that a simple content based technique [18] is as effective for this scenario, when searching for common tokens in name-like properties[11] of named entities.

Handling duplicate URIs. Once the sets of duplicate or co-referent URIs are detected, the next step is to determine how to resolve the co-reference and integrate information. A typical solution adopted in Linked Data is to link duplicates with `owl:sameAs` axioms. Glaser et al. [5] argue that `owl:sameAs` axioms are not suitable for co-referent URIs as they become indistinguishable, even though they may refer to different entities according to the context in which they are used. The Identity of Resources on the Web (IRW) ontology [6] formalises the distinction between information resources, such as webpages and multimedia files, and other kinds of Semantic Web non-information resources used in Linked Data and proposes a solution to link them. The solution adopted by DBpedia [11] for dealing with co-referent URIs follows the intuition of [5,6]: mirroring the structure of Wikipedia, DBpedia stores co-referent URIs in a separate graph called `redirects` and uses the HTTP response code `303 See Other` in order to (i) deal with the distinction between information and non-information resources and (ii) to implement the HTTP dereferencing.

Our solution encompasses those in [5,6,11] to allow to: (i) harmonise co-referent URIs by identifying information and non-information resources, and (ii) rely on HTTP redirect for dereferencing. We also extend it by enabling redirect in SPARQL queries by means of query rewriting. This enables backward compatibility to external clients that might refer to a certain resource by using any co-referent URIs.

[11] http://xmlns.com/foaf/0.1/name.

3 Deduplication

Given a set of URIs $E = \{e_1, e_2, ...e_n\}$ representing entities of the same type, the goal of deduplication is to: (i) identify sets of duplicate URIs that refer to the same real-world entity. We will call such URIs in each set *'co-referent'* to each other; (ii) determine in each subset one URI to keep (to be called the *'target URI'*), while deprecating the others (to be called the *'duplicates'*) and consolidating RDF triples from the duplicates into the target URI. In this work, we focus on two entity types that are found to be the dominant source of duplicates: **PERSON (PER)**, and **ORGANISATION (ORG)**.

For (i), we develop a process that identifies potential co-referent URI pairs $\langle e_i, e_j \rangle$ from E and submits each pair to a binary classifier to predict if they are truly co-referent. For (ii), we develop heuristics to deprecate duplicate URIs from co-referent pairs. Arguably, the first objective can also be achieved through the use of heuristic thresholds or clustering techniques. However, thresholds are often data-dependent [21], while our motivation for a classification-based approach is two-fold. First, as discussed before, our practical scenarios typically contain an existing, de-duplicated reference dataset D (i.e., ScholarlyData), and another data graph D' is generated from time to time and is to be merged into D (in our problem formalisation, both D and D' will be part of E). Hence the question to be asked is often classification-based: given a URI from D', determine if there exists one URI from D that refers to the same entity. Second, due to the nature of datasets, we expect clusters with 2 or more elements to be rather infrequent.

Beginning with an input set E, we first apply **blocking strategies** to identify pairs of URIs that are potentially co-referent. This should reduce the number of pairs to a number m such that $m \ll \binom{n}{2}$, where n is the number of URIs in E, $\binom{n}{2}$ is the number of *all* possible un-ordered pairs from E and will contain overwhelmingly negative elements as we expect true co-referent pairs to be rare. Next, each pair is passed to a **classification** process that predicts if the potential co-referent pair is positive. Finally, the positive co-referent pairs are submitted to the **URI harmonisation** process that identifies the target URI, removes the duplicates and merges RDF triples.

3.1 Blocking Strategies

Given the set of URIs E, the set P of all possible pair comparisons to perform is quadratic to the size of E. The blocking strategies aim at identifying a subset of comparisons P' such that $|P'| \ll |P|$, which achieves a good tradeoff between Reduction Ratio (RR) - the percentage of discarded comparisons from P - and Pair Completeness (PC) - the percentage of true positive (given a gold standard) that are covered by P'.

We experiment with two different solutions for this problem. First we use SNM [7]. We produce the list of all URIs in E with lexicographic ordering and we produce all combination for e_i with all e_j in a context window size of n (sliding window). We experiment with two kinds of ordering: one on the URIs

Table 1. Features for representing an organisation. $^{-1}$ indicates the inverse of a predicate.

Feature	Path	Target(s)	Example
Name	⟨conf:name⟩	Literals	'STLab-CNR', 'ISTC-CNR'
Members' names	⟨conf:withOrganisation^{-1}, conf:isAffliationOf, conf:name⟩	Literals	'L. Page', 'S. Brin'
Members' URIs	⟨conf:withOrganisation^{-1}, conf:isAffliationOf⟩	URIs	sdp:aldo-gangemi
Participated event URIs	⟨conf:withOrganisation^{-1}, conf:during⟩	URIs	sde:ESWC2009/ eswc/2009

themselves, one based on the literal value given the URI's predicate `conf`[12]`:name` for each e_i.

The second is a content based technique [18]. For each $e_i \in E$ we produce a candidate pair for comparisons for all e_j ($e_j \neq e_i$) that share at least one common token in their property values. Following [18], we choose name-like properties of entities (both persons and organisations). Details about experiments on the two blocking strategies are presented in Sect. 4.2.

3.2 Classification

Given a pair of URIs $\langle e_i, e_j \rangle$, we firstly build feature representations of e_i and e_j by traversing corresponding linked data graphs to gather their properties. Next, we derive a feature vector representation for the pair, which is then to be classified as either co-referent or not.

Features of URIs. These are generated by traversing paths on the linked data graph O, starting from e and following a series of predicates $\langle p_1, p_2, ...p_m \rangle$ that represent a particular semantic relation r, to reach another set of nodes, which can be either URIs or data literals. Depending on the semantic type of e, we define different semantic relations and paths. We use the ScholarlyData SPARQL endpoint[13] as the single point of access to the underlying linked data graph.

[12] The prefixes used in this paper are:

- `conf:` https://w3id.org/scholarlydata/ontology/conference-ontology.owl
- `sdp:` https://w3id.org/scholarlydata/person/
- `sde:` https://w3id.org/scholarlydata/event/
- `sdo:` https://w3id.org/scholarlydata/organisation
- `sdi:` http://www.scholarlydata.org/inproceedings/.

[13] https://w3id.org/scholarlydata/sparql/.

Table 2. Features for representing a person. The superscript $^{-1}$ indicates the inverse of a predicate.

Feature	Path	Target(s)	Example
Name	⟨conf:name⟩	Literals	'Tom Mitchell', 'T. Mitchell'
Affiliation names	⟨conf:hasAffiliation, conf:withOrganisation, conf:name⟩	Literals	'STLab-CNR', 'ISTC-CNR'
Affiliation URIs	⟨conf:hasAffiliation, conf:withOrganisation⟩	URIs	sdo:cnr-istc-italy
Participated event URIs	⟨conf:hasAffiliation, conf:during⟩	URIs	sde:ESWC2009/ eswc/2009
Published work URIs	⟨conf:hasContent^{-1}, conf:hasItem^{-1}, conf:hasAuthorList^{-1}⟩	URIs	sdi:ekaw2012/ paper/demos/109
Co-author URIs	⟨conf:hasContent^{-1}, conf:hasItem^{-1}, conf:hasItem, conf:hasContent⟩	URIs	sdp:aldo-gangemi, sdp:valentina-presutti
Title + abstract + keywords	starting from each each 'published work URI': ⟨conf:title⟩ ∨ ⟨conf:abstract⟩ ∨ ⟨conf:keyword⟩	Literals	'This paper describes a ...'

Table 3. Functions to measure similarity between two bag of features.

$$dice(e_i, e_j, t) = \frac{|f(e_i,t) \cap f(e_j,t)|}{|f(e_i,t) \cup f(e_j,t)|} \quad (1)$$

$$dice^{sqrt}(e_i, e_j, t) = \sqrt{sim_{dice}(f(e_i,t), f(e_j,t))} \quad (2)$$

$$cov(e_i, e_j, t) = \max\{\frac{|f(e_i,t) \cap f(e_j,t)|}{|f(e_i,t)|}, \frac{|f(e_i,t) \cap f(e_j,t)|}{|f(e_j,t)|}\} \quad (3)$$

$$cov^{sqrt}(e_i, e_j, t) = \sqrt{cov(e_i, e_j, t)} \quad (4)$$

For an ORG, we gather features following the paths shown in Table 1. We then normalise URI values to lowercase, and normalise literal values by ASCII folding and replacing any consecutive non-alphanumeric characters with a single white space. For a PER, we gather features following the paths shown in Table 2. We apply the same normalisation to URI and literal values as that for ORG. Furthermore, for the feature 'title+abstract+keyword', we also tokenise the text and remove stopwords.

Features of a Pair of URIs. Next, given a pair of URIs, we create a vector representation of the pair based on the similarities between the features of each URI. Let $f(e,t)$ return a bag of features for the feature type t of the URI e. Depending on t, this will be either a duplicate-removed set ($fs(e,t)$) or a multiset ($fm(e,t)$). We then measure the similarity of each feature type for a pair

of URIs in the range of $[0, 1]$, using the functions shown in Table 3. Functions based on the Dice co-efficient ($dice$, $dice^{sqrt}$) evaluate the extent to which the information that the two URIs have in common (the top term) along a particular dimension (feature type t), can describe both of them (the bottom term). Functions based on the Coverage (cov, cov^{sqrt}) evaluate the maximum degree to which the common part of the two can describe either of them. This is to cope with the situation where one URI is used much more frequently than the other, which is found to be common in linked datasets [21]. As a result, conventional set-overlap based measures (e.g., Dice) tend to produce low similarity scores for such pairs, due to the lack of features for the minor URI.

Specifically, for a pair of ORG URIs, we use $fs(e, t)$ as the bag of feature function, and apply similarity functions (1) to (4) to each of the four feature types in Table 1. This produces a feature vector with a dimension of 16. For a pair of PER URIs, we use all similarity functions with the $fs(e, t)$ bag of feature representation on all feature types (Table 2), except 'title+abstract+keyword', for which we use all similarities with the $fm(e, t)$ bag of feature representation). This produces a feature vector for the pair with a dimension of 28.

Classification Models. Given a labelled dataset of PER or ORG URI pairs, each represented as a feature vector described above, we train a binary classifier for predicting new, unseen PER or ORG pairs. A plethora of classification models can be used for this kind of tasks. In this work, we select five models to experiment with, including: a Stochastic Gradient Descent (SGD) classifier, a Logistic Regression (LR) model, a Random Forest (RF) decision tree model, a linear Support Vector Machine (SVM-l) and a nonlinear SVM using a Radial Basis Function kernel (SVM-rbf). We use the implementation from the scikit-learn[14] library for all models. Details of the datasets and the training process are discussed in Sect. 4.3.

3.3 URI Harmonisation

The previous steps allow to identify a set of pairs consisting of duplicate URIs in the ScholarlyData dataset. The harmonisation is the task of identifying which are the URIs to keep after the data cleansing process. This task is not trivial, as it requires multiple activities: (i) closure identification; (ii) candidate selection; (iii) knowledge inheriting; (iv) recording of the harmonisation.

Closure Identification. Here we traverse the transitive chains of duplicates in order to identify the closures of duplicate URIs from available pairs. Closure identification is mandatory as the classification process returns predictions on potential co-referent pairs. Hence, it is possible to have transitive chains of co-referent pairs resulting from the classification in case a certain person or organisation is represented by more than two URIs in the dataset. This scenario

[14] http://scikit-learn.org/.

occurs when the lexicalisations provided for the names of people and organisation vary from one conference to another. For example, the individual 'Andrea Giovanni Nuzzolese' is associated with the URIs `sdp:andrea-giovanni-nuzzolese`, `sdp:andrea-nuzzolese`, and `sdp:andrea-g-nuzzolese`, which in turn are provided by the classifier as a set of pairs as follows.

```
<sdp:andrea-nuzzolese,sdp:andrea-giovanni-nuzzolese>
<sdp:andrea-g-nuzzolese,sdp:andrea-nuzzolese>
<sdp:andrea-giovanni-nuzzolese,sdp:andrea-g-nuzzolese>
```

Closure identification then produces a single tuple as:

```
<sdp:andrea-nuzzolese,sdp:andrea-giovanni-nuzzolese,sdp:andrea-g-nuzzolese>
```

Candidate Selection. The result of the previous activity is a set of tuples. Next, candidate selection aims at identifying for each tuple, a single candidate to keep as reference URI. The selection is performed by computing and comparing the degree centrality of each entity associated with a URI part of a tuple. The degree centrality is computed as the sum of indegree and outdegree, which count the number of incoming and outcoming ties of the node respectively. Accordingly, the selected URI is the one recording the highest degree centrality within the context of the tuple. To compute outdegree centrality, we do not take into account of datatype properties, as in ScholarlyData the only literals used for person and organisation are names. Hence, we assume that (i) an entity with alternative names is not more relevant with respect to another with a single name; (ii) the centrality is captured by object properties only.

Dataset Update. Once a single URI for each tuple has been selected, we associate the triples available through the other URIs of the tuple with the single selected URI. This is performed by means of SPARQL UPDATE queries.

Recording of the Harmonisation. This is carried out in parallel with the previous activity and its result is an RDF graph whose aim is twofold: (i) providing a description about how a final URI for a set of duplicates is derived and (ii) serving as background knowledge for enabling backward compatibility. The backward compatibility is needed in order to guarantee transparent access to any client application that relies on ScholarlyData without any client-side change being required. The RDF graph is modelled by using PROV-O[15] and describes the harmonisation in terms of provenance. Following the previous example, we show the resulting RDF graph generated below.

```
sdp:andrea-giovanni-nuzzolese a prov:Entity, sdo:Person ;
  prov:wasDerivedFrom sdp:andrea-nuzzolese, sdp:andrea-g-nuzzolese .
sdp:andrea-nuzzolese a prov:Entity, sdo:Person .
sdp:andrea-g-nuzzolese a prov:Entity, sdo:Person .
```

[15] https://www.w3.org/TR/prov-o/.

In this example, the entity `sdp:andrea-giovanni-nuzzolese` is typed as `prov:Entity` and associated with the entities `sdp:andrea-nuzzolese` and `sdp:andrea-g-nuzzolese` by means of the property `prov:wasDerivedFrom` that according to PROV-O allows to model the transformation of an entity into another, an update of an entity resulting in a new one or the construction of a new entity based on a pre-existing entity [9]. The objects of `prov:wasDerivedFrom` properties are recorded in the RDF graph along with their original triples[16] coming from the dataset before the update. On top of the RDF graph resulting from this activity we designed and implemented a software module that enables HTTP redirect and SPARQL query expansion in case deleted URIs are requested. The HTTP redirect is implemented by querying the provenance graph when a resource is not available in the default graph of ScholarlyData. Hence, if such a resource is available in the provenance graph the software module return a 301 HTTP status code, meaning that the resource has been moved permanently. For example, of response a HTTP/1.1 301 Moved Permanently to `sdp:andrea-giovanni-nuzzolese` is returned when the resource `sdp:andrea-nuzzolese` is requested.

The SPARQL query expansion is implemented by substituting all the occurrences of deleted URIs that appear as URI constants in a query with the valid ones. Again this is enabled by the provenance graph. The software is available on the GIT repository of cLODg[17]. For example, in the following, the first SPARQL SELECT statement is automatically converted to the second and then executed assuming that `sdp:andrea-giovanni-nuzzolese` is the candidate URI selected to inherit the knowledge from its duplicates and `sdp:andrea-nuzzolese` is one of the removed duplicates.

```
SELECT DISTINCT ?pred ?obj WHERE {sdp:andrea-nuzzolese ?pred ?obj}
SELECT DISTINCT ?pred ?obj WHERE {sdp:andrea-giovanni-nuzzolese
?pred ?obj}
```

4 Experiments

4.1 The Train/test Dataset

We manually labelled a dataset of pairs of PER URIs (*perD*) and pairs of ORG URIs (*orgD*). To select the PER pairs to annotate we retrieve all pairs that share at least a common value in one of their properties (we restrict these to `conf:name`, `conf:familyName`, `rdfs:label`; note that these properties can return multiple values for each entity). Similarly we select the ORG pairs (where the properties are restricted to `conf:name`, `rdfs:label`). For ORG we generate additional pairs by retrieving all affiliations for each person and generating all pair combinations when multiple affiliations are found for each person. We then manually annotated roughly 20% of all the candidates as positive or negative, discarding all the others. This resulted in 698 (148 positive, 550 negative) pairs for *perD* and 424 (188 positive, 236 negative) pairs for *orgD*.

[16] Due to space limitation those triples have been omitted in the example.
[17] https://github.com/anuzzolese/cLODg2.

4.2 Blocking

This part of experiment is aimed at identifying the optimal strategy for reducing pair comparisons. We evaluate this independently from the classification step by considering the *pair completeness* (PC), i.e. the fraction of true positive entity pairs contained in the list of candidate pairs returned by blocking, and the *reduction ratio* (RR), i.e. the percentage of pairs discarded after blocking is applied from the total number of possible pairs without blocking. The *harmonic mean* (HM) measures the tradeoff between PC and RR. These metrics give us some indication of the upper bound of performance for the classifiers, i.e. the percentage of duplicates that can be potentially identified, assuming perfect prediction. Given all PER or ORG URIs from ScholarlyData, we first run different blocking strategies to create candidate URI pairs. Next, we use the true positive pairs from the training data as reference to calculate the three metrics. Note that it is expected that this reference set may not include all true positives in the entire ScholarlyData dataset.

Tables 4 and 5 show the results of the blocking strategies for PER and ORG respectively. For the content based method, we keep all URI pairs that share at least a common value in their name-like properties. For the SNM method we produce the list of all URIs with the two orderings described in Sect. 3.1 and for each URI we generate all combinations in a context window n, from 5 to 90.

Results show that the content based method produces the best results for both PER and ORG. Note that although in training data preparation, we also used name-like properties as a proxy to find candidate pairs to annotate, the content based blocking uses name-like properties in a more general way (i.e., we tokenise property values and look for common tokens rather than values) and in practice generates far more candidate pairs.

We conclude to use the content based blocking for both PER and ORG to create candidate URI pairs. Specifically, we use multiple features L+N+S for PER (ref. Table 4) and P+N+L for ORG (ref. Table 5). We choose multiple

Table 4. Blocking results for PER. Results for (i) the SNM method (ordered by URI/name), at the variation of the window size, and (ii) the content based method with different features: N stands for `conf:name`, S for `conf:familyName`, L for `rdfs:label`.

SNM name	RR	PC	HM	SNM URI	RR	PC	HM	Content based	RR	PC	HM
5	≈ 1	0.73	0.84	5	≈ 1	0.6	0.75	L	≈ 1	0.38	0.55
10	≈ 1	0.8	0.89	10	≈ 1	0.71	0.83	N	≈ 1	0.38	0.55
20	≈ 1	0.91	0.95	20	≈ 1	0.8	0.89	S	≈ 1	1	≈ 1
30	0.99	0.91	0.95	30	0.99	0.84	0.91	L+N+S	≈ 1	1	≈ 1
50	0.99	0.93	0.96	50	0.99	0.84	0.91				
70	0.99	0.93	0.96	70	0.99	0.85	0.91				
90	0.98	0.95	0.97	90	0.98	0.89	0.93				

Table 5. Blocking results for ORG. Including: (i) the SNM method (ordered by URI/name), at the variation of the window size, and (ii) the content based method with different features: N stands for `conf:name`, P for affiliated person, L for `rdfs:label`.

SNM name	RR	PC	HM	SNM URI	RR	PC	HM	Content based	RR	PC	HM
5	≈ 1	1	≈ 1	5	≈ 1	0.49	0.66	P	≈ 1	0.84	0.91
10	0.99	1	≈ 1	10	0.99	0.59	0.74	N	≈ 1	1	≈ 1
20	0.99	1	0.99	20	0.99	0.65	0.78	L	≈ 1	1	≈ 1
30	0.98	1	0.99	30	0.98	0.66	0.79	P+N+L	≈ 1	1	≈ 1
50	0.97	1	0.99	50	0.97	0.68	0.8				
70	0.96	1	0.98	70	0.96	0.68	0.8				
90	0.95	1	0.98	90	0.95	0.71	0.81				

features over single features (i.e., S for PER, N or L for ORG) because in practice, using multiple features for content based blocking may result in better PC. Although the effects of using multiple as opposed to single features are indifferent for the experiment results, this may be partially attributed to the potentially incomplete reference set of true positive pairs, as discussed above.

Applied to the entire sections of PER and ORG URIs from the ScholarlyData dataset, this generates 1,468 PER and 3,717 ORG URIs to be classified, which is around 1% of the total possible pair combinations, and we expect this reasonably leads to a significant reduction in running time.

4.3 Classification

This part of experiment evaluates the performance of the binary classification models, using the datasets created in Sect. 4.1. The task addresses a typical problem in link discovery, therefore we compare against two state-of-the-art systems to be described below.

Training and testing. We split each labelled dataset (*perD* and *orgD*) randomly into a training set containing 75% of the data, and a testing set containing 25% of data. For each of our classifiers, we tune their hyper parameters by performing grid search using the training set with 10-fold cross validation. Both the labelled datasets and the optimised classification models are available for download[18]. The optimised classifiers are then applied to the test set. Both PER and ORG experiments are carried out independently from each other.

State-of-the-art. We compare our models against LIMES[19] and SILK[20], both of which offer supervised learning methods that can benefit from training data.

[18] https://github.com/ziqizhang/scholarlydata/tree/master/data/public/.
[19] Ver 1.1.2, https://github.com/AKSW/LIMES-dev/releases.
[20] Ver 2.6, https://github.com/silk-framework/silk/releases.

Table 6. Classification results for ORG.

		SGD	LR	RF	SVM-l	SVM-rbf
Positive examples	P	0.85	0.84	0.86	0.83	0.83
	R	0.8	0.8	0.82	0.86	0.86
	F1	0.83	0.82	0.84	0.85	0.85
Negative examples	P	0.83	0.82	0.84	0.87	0.87
	R	0.87	0.85	0.87	0.84	0.84
	F1	0.85	0.84	0.86	0.85	0.85
Total	P	0.84	0.83	0.85	0.85	0.85
	R	0.84	0.83	0.85	0.85	0.85
	F1	0.84	0.83	0.85	0.85	0.85

Table 7. Classification results for PER.

		SGD	LR	RF	SVM-l	SVM-rbf
Positive examples	P	0.78	0.88	0.92	0.77	0.88
	R	0.51	0.63	0.66	0.49	0.63
	F1	0.62	0.73	0.77	0.6	0.73
Negative examples	P	0.89	0.91	0.92	0.88	0.91
	R	0.96	0.98	0.99	0.96	0.98
	F1	0.93	0.94	0.95	0.92	0.94
Total	P	0.87	0.91	0.92	0.88	0.91
	R	0.87	0.91	0.92	0.87	0.91
	F1	0.86	0.90	0.91	0.86	0.90

For LIMES, we test *Wombat Simple* (**L-ws**) and *Wombat Complete* (**L-wc**) *supervised batch* models that are available at the time of writing. For a consistent experimental environment, the same set of URI features (Sect. 3.2) are used for all models. Both SILK and LIMES models are trained and tested on the train-test splits (75%–25%) for ORG and PER respectively. Their default configurations are used.

Analysis. Tables 6 and 7 show results of our five classifiers. The results show that detecting and resolving duplicates in scholarly linked datasets is not an easy task. When only positive examples are considered - as in the realistic scenarios, the performance of the classifiers is particularly weak for PER in terms of recall. Manual inspection of the training datasets reveals that, *on the one hand*, features of some URIs are very sparse. For example, URI `sdp:gregoris-antoniou` has no published work, while `sdp:ghislain-atemezing` has no affiliations. Consequently, the similarity between this URI and its true positive co-referent URI will be 0 in terms of this feature. To rectify this, an ensemble of classifiers could be employed. Different classifiers can be trained on different sub-sets of feature types. Then during testing, the optimal model is chosen dynamically depending on the availability of feature types. *On the other hand*, two URIs in a pair sometimes use features that, although are disjoint, often share certain implicit connections. For example, conference participants may use affiliations of different granularity from time to time, e.g., the names 'The Open University' and 'KMI'. Being able to measure the implicit connectivity between the two values could improve the model's recall. *Furthermore*, to improve recall, generalisation over certain features could also be helpful. For example, we derive generic event names (e.g., ESWC) based on their URIs, currently representing event series (e.g., ESWC2009, ESWC2011). All of these will be explored in the future work.

Overall, the RF model offers the best trade-off for both PER and ORG, especially on positive examples. Therefore in Tables 8 and 9 we compare the

Table 8. SoA results for ORG.

		RF	L-ws	L-wc	SILK
Positive examples	P	0.86	0.63	0.73	0.59
	R	0.82	0.64	0.49	1.0
	F1	0.84	0.64	0.59	0.74
Negative examples	P	0.84	0.72	0.69	1.0
	R	0.87	0.71	0.86	0.47
	F1	0.86	0.72	0.77	0.64
Total	P	0.85	0.68	0.70	0.70
	R	0.85	0.68	0.70	0.70
	F1	0.85	0.68	0.70	0.70

Table 9. SoA results for PER.

		RF	L-ws	L-wc	SILK
Positive examples	P	0.92	0.34	0.78	0.48
	R	0.66	0.63	0.17	0.80
	F1	0.77	0.44	0.28	0.6
Negative examples	P	0.92	0.85	0.79	0.94
	R	0.99	0.61	0.98	0.71
	F1	0.95	0.71	0.88	0.81
Total	P	0.92	0.62	0.79	0.79
	R	0.92	0.62	0.79	0.73
	F1	0.91	0.62	0.79	0.76

results of this model against the best results of SILK and LIMES models[21]. As the results show, our method makes significant improvement in both tasks on F1, and strikes much better balance between precision and recall. This could be partially attributed to the usage of the 'Coverage' functions that may cope with infrequently used URIs more effectively. However, readers should note that on the one hand, our method is specifically tailored to the problem while both LIMES and SILK are general purpose matching systems; on the other hand, we did not extensively test all configurations of the two systems but used their default settings.

Finally, we use the trained RF model to label the sets of PER and ORG URI pairs identified by content based blocking, as discussed before. The output is then passed to URI harmonisation.

4.4 URI Harmonisation

The aim of this experiment is twofold: (i) assessing the classification output, and (ii) updating the dataset. First we manually checked all pairs labelled as positive. For PER we recorded 101 correct resolutions over 118, i.e. 0.86 of precision. For ORG we recorded 884 correct resolutions out of 1,262 pairs, i.e. 0.7 of precision. We then used the resulting cleaned output to harmonise the URIs on the ScholarlyData dataset. Consequently, from 101 pairs of PER URIs we derived 94 unique individuals each of them harmonising on average 2.05 distinct individuals. The average size of the graph associated with each individual

[21] LIMES allows setting a threshold for predicted mappings. We tested different thresholds from 0.1 to 1.0 with increment of 0.1 and found that LIMES-wc is insensitive to the threshold while LIMES-ws is. For complete results and optimal thresholds see: https://github.com/ziqizhang/scholarlydata/tree/master/data/public/soa_results.

involved in the harmonisation for PER counted of 20.3 distinct RDF triples. Similarly, for ORG we kept 531 correct resolutions out of 884 pairs harmonising 2.67 URIs on average. The graph associated with each individual involved in the harmonisation for ORG counted of 10.2 distinct RDF triples on average.

5 Conclusions

In this work, we introduced an approach to address a key issue in the publication of scholarly linked data on the Semantic Web, i.e., the presence of duplicate URIs for the same entities. Using the ScholarlyData dataset as a reference, our approach uses *blocking techniques* to narrow down a list of candidate duplicate URI pairs, exploits *supervised classification methods* to label the true positives and then devises a protocol to choose the most representative URI for an entity to keep in ScholarlyData and to make sure that we preserve all facts from duplicated URIs. To our knowledge, this is by far the first attempt to solve such issues on the largest conference dataset in the Semantic Web community. Future work will be carried out in a number of directions. Firstly, we will look into the issue of other types of URIs, such as events. Next, in terms of the classification process, we will explore the possibilities of improvement discussed before. We will also develop methods that exploit the dependency between the different types of URIs, where the solution to one task can feed into that of another (e.g., the de-duplication of ORG URIs could potentially address the disjoining issue of 'affiliation URI' and 'affiliation names' features of PER). Finally, we will explore the inclusion of human in the loop, in an 'active-learning' fashion to both minimise the human effort on annotation and improve the accuracy of our method.

References

1. Bryl, V., Birukou, A., Eckert, K., Kessler, M.: What is in the proceedings? combining publishers and researchers perspectives. In: SePublica 2014 (2014)
2. Clark, K., Manning, C.: Entity-centric coreference resolution with model stacking. In: Association for Computational Linguistics (2015)
3. Duan, S., Fokoue, A., Hassanzadeh, O.: Instance-Based Matching of Large Ontologies Using Locality-Sensitive Hashing. pp. 49–64 (2012)
4. Gentile, A.L., Acosta, M., Costabello, L., Nuzzolese, A.G., Presutti, V., Reforgiato Recupero, D.: Conference live: accessible and sociable conference semantic data. In: Proceedings of WWW Companion, pp. 1007–1012 (2015)
5. Glaser, H., Jaffri, A., Millard, I.: Managing co-reference on the semantic web. In: Linked Data on the Web (LDOW 2009) (2009)
6. Halpin, H., Presutti, V.: The identity of resources on the web: an ontology for web architecture. Appl. Ontol. **6**(3), 263–293 (2011)
7. Hernandez, M.A., Stolfo, S.J.: The merge/purge problem for large databases. In: Proceedings of SIGMOD 1995. ACM (1995)
8. Isele, R., Bizer, C.: Learning expressive linkage rules using genetic programming. Proc. VLDB Endow. **5**(11), 1638–1649 (2012)
9. Lebo, T., Sahoo, S., McGuinness, D.: Prov-o: The prov ontology. W3C recommendation, W3C, April 2013. https://www.w3.org/TR/prov-o/

10. Lee, D., Kang, J., Mitra, P., Giles, C.L., On, B.-W.: Are your citations. Commun. ACM **50**(12), 33–38 (2007)
11. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia. Semant. Web J. **6**, 167–195 (2013)
12. Mamun, A.-A., Aseltine, R., Rajasekaran, S.: Efficient record linkage algorithms using complete linkage clustering. PLoS ONE **11**(4), e0154446 (2016)
13. Möller, K., Heath, T., Handschuh, S., Domingue, J.: Recipes for semantic web dog food — the ESWC and ISWC metadata projects. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC/ISWC - 2007. LNCS, vol. 4825, pp. 802–815. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76298-0_58
14. Nentwig, M., Hartung, M., Ngomo, A.-C.N., Rahm, E.: A survey of current link discovery frameworks. Semant. Web (Preprint):1–18 (2015)
15. Ngomo, A.-C.N., Auer, S.: LIMES: a time-efficient approach for large-scale link discovery on the web of data. In: Proceedings of IJCAI 2011, pp. 2312–2317 (2011)
16. Nuzzolese, A.G., Gentile, A.L., Presutti, V., Gangemi, A.: Conference Linked data: the scholarlydata project. In: Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y. (eds.) ISWC 2016. LNCS, vol. 9982, pp. 150–158. Springer, Cham (2016). doi:10.1007/978-3-319-46547-0_16
17. Osborne, F., Motta, E., Mulholland, P.: Exploring scholarly data with rexplore. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) ISWC 2013. LNCS, vol. 8218, pp. 460–477. Springer, Heidelberg (2013). doi:10.1007/978-3-642-41335-3_29
18. Papadakis, G., Niederée, C., Fankhauser, P.: Efficient entity resolution for large heterogeneous information spaces. pp. 535–544 (2011)
19. Shotton, D.: Semantic publishing: the coming revolution in scientific journal publishing. Learned Publishing **22**(2), 85–94 (2009)
20. Solecki, B., Silva, L., Efimov, D.: KDD cup 2013: author disambiguation. In: Proceedings of the 2013 KDD Cup 2013 Workshop, KDD Cup 2013, pp. 9:1–9:3. ACM, New York (2013)
21. Zhang, Z., Gentile, A.L., Blomqvist, E., Augenstein, I., Ciravegna, F.: An unsupervised data-driven method to discover equivalent relations in large linked datasets. Semant. web **8**(2), 197–223 (2017)
22. Zheng, J., Chapman, W., Crowley, R., Savova, G.: Coreference resolution: a review of general methodologies and applications in the clinical domain. Biomed. Inform. **44**(6), 1113–1122 (2011)