Selecting Suitable Configurations for Automated Link Discovery

Carlos R. Rivero
crr@cs.rit.edu
Rochester Institute of Technology, USA

David Ruiz
druiz@us.es
University of Seville, Spain

ABSTRACT

Linking individuals in one dataset to other same individuals in existing datasets is a major problem known as link discovery. Existing automated link discovery techniques make users responsible for selecting suitable properties, distances and transformations, a.k.a. configurations, which is challenging for both researchers and practitioners. Furthermore, failing to provide suitable configurations dramatically increases the complexity of link discovery since many configurations need to be evaluated. Current approaches to help users select proper configurations assume datasets are not heterogeneous or require the existence of a schema or ontology, making them less appealing in the context of Linked Data. In this paper, we present an approach to help users select suitable configurations solely based on data, i.e., no schema or ontology is required. We rely on the concepts of universality and uniqueness, i.e., properties that are present in many individuals of the datasets to link (universality) and do not have repeated objects (uniqueness). We use the concept of singularity to focus on configurations in which only a few individuals are very similar while the rest are very dissimilar. We evaluate our approach using eight commonly-used scenarios, in which, on average, we only suggest 5% of all the possible configurations. Additionally, selected configurations consistently generate links achieving high precision and recall with respect to a ground truth. Finally, we provide a number of guidelines to apply our approach in additional scenarios.

KEYWORDS

Linked data, link discovery, data integration

1 INTRODUCTION

Link discovery aims to identify and link individuals belonging to two datasets that are the same, e.g., the same restaurant is referred to as "Art's Delicatessen" in a dataset and as "Art's Deli" in another dataset [1, 23]. Link discovery is crucial when publishing Linked Data to connect to other existing individuals [5, 14, 29]. Link discovery is very related to the problem of entity resolution that consists of detecting duplicate individuals in datasets [7, 21].

Current techniques to automatically perform link discovery usually rely on rules that are applied over the source and target datasets to be linked [21], e.g., a pair of restaurants is linked when the similarity between their names is below 0.35 using the cosine string distance. Genetic algorithms are appealing to compute these link discovery rules [10, 17, 18, 24, 27]. Furthermore, there are other techniques not based on rules to discover links that rely on machine learning [31]. The main difference between both types is that rules can be interpreted, refined and combined [8], while classifiers learned using machine learning are generally black boxes.

The search space that automated link discovery techniques need to traverse includes the following:

- A variety of properties present in the source and target datasets to be linked.
- Distances to compare the objects of these properties.
- Transformations to be applied over these objects.
- Thresholds to whether or not consider a certain object comparison as similar.
- Aggregate functions that allow to combine multiple of these constructs.

In the context of RDF datasets, distances and transformations are usually mandatory since the datasets tend to be heterogeneous [5]. In addition, the quality of the links output by these techniques largely depends on suitable combinations of properties, distances and transformations, thresholds, and aggregate functions.

Many automated link discovery techniques make users responsible for selecting suitable sets of source and target properties, and which pairs of source and target properties are appealing to be compared [10, 17, 18, 24, 27, 31]. Users also need to provide a number of distances and transformations to achieve such comparisons. This is a valid approach when users are very familiar with the source and target datasets to be linked and, therefore, can usually provide useful information to the techniques. However, for datasets that have not been previously analyzed, users are forced to either provide a wide range of options, which makes the problem of finding links computationally expensive or even unfeasible, or to take wild guesses in a trial-and-error manner until discovering plausible configurations. A configuration consists of a pair of source and target properties along with a distance, a pair of source and target transformations, and a threshold. Note that these techniques are thus responsible for finding proper single or aggregations of configurations as well as thresholds to discover links.

Several approaches to help select suitable configurations assume the same objects are present in the source and target datasets and, therefore, do not consider distances and/or transformations [3, 25].

Other approaches require to evaluate every pair of source and target individuals to discern whether they form a link [11], which may not be feasible for large datasets, or require the existence of a schema or ontology [2, 4, 13, 20, 26–28, 33], which is a strong requirement in the context of Linked Data since data models usually contain very few or no constraints at all [16].

In this paper, we present an approach to help users select suitable configurations for link discovery that solely uses the datasets involved without requiring the existence of any ontology or schema. These configurations can then be used by any automated link discovery technique to refine thresholds and/or be aggregated to effectively generate links. We rely on the concepts of universality, uniqueness and singularity. For a given dataset, universality of a property measures the percentage of individuals in the dataset that are related to any objects by means of such a property. A large universality score suggests a property is pervasive among individuals of a given dataset, e.g., all individuals have a name. Uniqueness measures the percentage of unique objects in the dataset that are related to individuals by means of such a property. A large uniqueness score entails a property is always related to unique objects, e.g., the address of a restaurant. Having a source and a target datasets, singularity measures the information gain of the individuals related to a given pair of source and target properties that are present in the links using a distance, a pair of transformations and a threshold. A large singularity score implies that individuals are very similar in a very few number of cases and very dissimilar in the rest. For instance, the names of restaurants "Art's Delicatessen" and "Art's Deli" are very similar (few) but very dissimilar to other (many) names, such as "Montrachet," "Spago," or "Tolouse."

Similar concepts related to universality and uniqueness have been proposed [4, 15, 25]; however, they either require the materialization of links to recommend promising source and/or target properties [4, 15], or they are combined using the harmonic mean that does not prioritize uniqueness with respect to universality [25], which is generally desired in the context of link discovery. To the best of our knowledge, none of the existing approaches suggest promising configurations to discover links. We evaluate our approach in eight scenarios that have been previously used in the literature [21, 23], in which we show that our approach recommends a set of configurations that achieve high precision and recall with respect to all possible configurations. Furthermore, we are able to reduce the search space of automated link discovery techniques in 95% on average. We also provide a number of guidelines to help users exploit universality, uniqueness and singularity scores in third-party scenarios. Finally, we make our implementation and experimental results publicly available to ensure reproducibility[1].

This paper is organized as follows: Section 2 deals with preliminaries; Section 3 discusses our approach; Section 4 reports our experimental results; Section 5 presents the related work; and Section 6 recaps our conclusions.

2 PRELIMINARIES

A triple (i, p, o) relates an individual i to an object o by means of property p: i can be an IRI or a blank node, p can only be an IRI, and o can be either an IRI, a blank node or a literal [9]. A RDF dataset D is formed by a set of triples. We denote the set of individuals in D as $I(D) = \{i | (i, p, o) \in D\}$ and the set of properties as $P(D) = \{p | (i, p, o) \in D\}$. Note that we do not consider individuals that appear only as objects, which entails that these individuals are not related to any objects by means of datatype properties, i.e., similar to attributes in relational databases. In practice, our approach can deal with these individuals if the inverse of existing object properties are also considered.

(id_1, *name*, '*toulouse*')	(id_A, *fullname*, '*Toulouse*')
(id_1, *category*, '*French cuisine*')	(id_A, *type*, '*French*')
(id_2, *name*, '*hotel bel–air*')	(id_B, *fullname*, '*Spago*')
(id_3, *name*, '*Art's Delicatessen*')	(id_B, *type*, '*French*')
(id_3, *category*, '*French cuisine*')	(id_C, *fullname*, '*Art's Deli*')
(id_4, *name*, '*le montrachet*')	(id_D, *fullname*, '*Montrachet*')
(id_4, *name*, '*montrachet*')	(id_D, *type*, '*French*')
(a) Source (R_S)	**(b) Target (R_T)**

Figure 1: Sample datasets representing restaurants

Example 2.1. Figure 1a presents a dataset R_S containing restaurants where $I(R_S) = \{id_1, id_2, id_3, id_4\}$ and $P(R_S) = \{name, category\}$. Note that, for the sake of simplicity, we present string literals like '*toulouse*' and we assume that the rest are IRIs like id_1 or *name*.

Let D_S and D_T be two datasets, a link is a pair of individuals (i_s, i_t) where $i_s \in I(D_S)$ and $i_t \in I(D_T)$. A configuration is a tuple $(p_s, p_t, \psi, \delta_s, \delta_t, \theta)$ where $p_s \in P(D_S)$ and $p_t \in P(D_T)$, ψ is a distance, δ_s and δ_t are transformations, and $\theta \in [0, 1]$ is a threshold. A configuration may be used to generate a link (i_s, i_t) as follows: Assuming $(i_s, p_s, o_s) \in D_S$ and $(i_t, p_t, o_t) \in D_T$, (i_s, i_t) is created if $\psi(\delta_s(o_s), \delta_t(o_t)) \leq \theta$. Note that we assume distances are normalized and produce a value between 0 and 1; the former implies both objects are considered identical. In practice, a single configuration is usually not enough for discovering links, so automated link discovery techniques focus on finding and aggregating configurations to generate proper links between datasets.

Example 2.2. Assuming that *Lev* represents the Levenshtein string distance and *Void* denotes no transformation, a configuration like (*name*, *fullname*, *Lev*, *Void*, *Void*, 0.35) produces links (id_1, id_A) and (id_4, id_D) in Figure 1 since the threshold is met as follows:

- $Lev(Void(\text{'}toulouse\text{'}), Void(\text{'}Toulouse\text{'})) = 0.13$
- $Lev(Void(\text{'}le\ montrachet\text{'}), Void(\text{'}Montrachet\text{'})) = 0.31$

Since $Lev(Void(\text{'}Art's\ Delicatessen\text{'}), Void(\text{'}Art's\ Deli\text{'})) = 0.44$, ($id_3$, id_C) is not output by the previous configuration when used to generate links between R_S and R_T.

3 OUR APPROACH

In this section, we introduce the concepts of universality and uniqueness that are applied over a single dataset to select promising individual properties (Section 3.1). We present the concept of singularity that is applied over pairs of source and target properties to select promising distances, transformations, and thresholds (Section 3.2).

[1] https://github.com/crrivero/CHALD

3.1 Selecting individual properties

We aim to help users select promising properties in a given dataset D, either the source or the target, based on universality and uniqueness. The universality score $u(p, D)$ of a property p is defined as:

$$u(p, D) = \frac{|E(p, D)|}{|I(D)|}$$

where $E(p, D) = \{i \mid (i, p, o) \in D\}$ is a set of unique individuals for property p. Note that $u(p, D) = 1$ entails that all individuals in D have at least an object for property p. Intuitively, a universality score $u(p, D) \approx 1$ implies that p is very prevalent among all individuals that are present in a given dataset.

Example 3.1. In Figure 1a, $u(name, R_S) = 4/4 = 1$ since all individuals have a triple for property *name*; $u(category, R_S) = 2/4 = 0.5$ since individuals id_2 and id_4 do not have any triples containing the *category* property.

The uniqueness score $q(p, D)$ of a property p is defined as:

$$q(p, D) = \frac{|V(p, D)|}{|R(p, D)|}$$

where $V(p, D) = \{o \mid (i, p, o) \in D\}$ is a set of unique objects for property p, and $R(p, D) = \{(i, o) \mid (i, p, o) \in D\}$ is the unique pairs individual–object for a given property p. Similarly as before, $q(p) = 1$ entails that all values related to individuals in D by means of property p are unique. Intuitively, a uniqueness score $q(p, D) \approx 1$ implies that the objects related by property p are mostly unique in a given dataset. Note that if $E(p, D)$ were used instead of $R(p, D)$, we could have $q(p, D) > 1$ in datasets where the same individual is related to different objects by the same property.

Example 3.2. In Figure 1a, $q(category, R_S) = 1/2 = 0.5$ since the dataset only contains value '*French cuisine*' for such property, and there are two individual–object pairs that have the *category* property. Furthermore, $q(name, R_S) = 5/5 = 1$ since there are five different names and there are five individual–object pairs related to such property. In Figure 1b, $q(type, R_T) = 1/3 = 0.33$ since the dataset only contains object '*French*', and three individual–object pairs have triples using property *type*.

In general, we are interested in properties whose uniqueness scores are high since the objects they relate are more likely to produce quality links. Assuming that the uniqueness score for a given property is high, the universality score indicates how likely it is to generate all links using a few or more linking rules.

For users to provide a single threshold, existing approaches have proposed to combine scores using the harmonic mean [4, 25]. We argue that the harmonic mean is only useful when both scores are equally high or low; otherwise, the harmonic mean does not allow to discard undesirable cases. For instance, for the *type* property in Figure 1b, we obtain $u(type, R_T) = 0.75$ and $q(type, R_T) = 0.33$ whose harmonic mean is equal to 0.46; link rules based on *type* are not expected to produce quality links. Assume a property p_1 in R_T such that $u(p_1, R_T) = 0.30$ and $q(p_1, R_T) = 0.95$ whose harmonic mean is also equal to 0.46; however, link rules based on p_1 are expected to produce quality links because of its high uniqueness.

We prioritize uniqueness with respect to universality and combine them using F_β as follows:

$$F_\beta(p, D) = \frac{(1 + \beta^2)\, u(p, D)\, q(p, D)}{(\beta^2\, u(p, D)) + q(p, D)}$$

where $\beta \geq 1$ in order to promote the uniqueness score, i.e., if $\beta < 1$, the universality score is promoted. In the previous example, $F_2(type, R_T) = 0.37$ while $F_2(p_1, R_T) = 0.66$, which allows us to differentiate between both properties.

3.2 Selecting configurations

Given a source and a target datasets D_S and D_T, a set of distances, and a set of transformations, we aim to find which combinations of source and target properties are appealing to be checked during the link discovery process, as well as proper distances, transformations and thresholds. We rely on the concept of singularity in which only a few and unique values are very similar, while the rest are very dissimilar taking distances and transformations into account. As a result, we find configurations, each of which is a tuple of the form $c = (p_s, p_t, \psi, \delta_s, \delta_t, \theta)$, whose singularity scores are above certain threshold. We compute singularity scores based on the links a given configuration generates between D_S and D_T.

For a given pair of triples (i_s, p_s, o_s) and (i_t, p_t, o_t) that belong to D_S and D_T, respectively, $\psi(\delta_s(o_s), \delta_t(o_t))$ is the distance between i_s and i_t for properties p_s and p_t. Having a threshold $\theta \in [0, 1]$, $L(c)$ is the set of links generated by configuration c as follows: $L(c) = \{(i_s, i_t) | (i_s, p_s, o_s) \in D_S \wedge (i_t, p_t, o_t) \in D_T \wedge \psi(\delta_s(o_s), \delta_t(o_t)) \leq \theta\}$. When $\theta = 0$, $L(c)$ contains all individuals for which the objects related by properties p_s and p_t are identical after applying δ_s and δ_t, and according to ψ. When $\theta = 1$, $L(c)$ contains the Cartesian product of all individuals that use properties p_s and p_t.

We focus on the number of unique source and target individuals that are present in the links generated by a given configuration. Thus, the singularity score $g(c)$ of a configuration c such that $|L(c)| > 1$ is defined as:

$$g(c) = \frac{H(D_S, L(c)) + H(D_T, L(c))}{2 \log |L(c)|}$$

where $H(D, L)$ represents the entropy of the individuals in D present in L computed as follows:

$$H(D, L) = -\sum_{i \in I(D)} \frac{|F(i, L)|}{|L|} \log \frac{|F(i, L)|}{|L|}$$

where $F(i, L) = \{(i_s, i_t) \mid (i_s, i_t) \in L \wedge (i = i_s \vee i = i_t)\}$ is the set of links in L that relate a given individual i.

The singularity score is computed based on the entropy of the source and target individuals present in $L(c)$. The intuition is that, if a source or a target individual is just present a few times in the links generated by configuration c, it constitutes a singularity, thus we expect a higher singularity score. Furthermore, we consider $g(c) = 1$ if $|L(c)| = 1$, and $g(c) = 0$ if $|L(c)| = 0$.

Example 3.3. In Figure 1, configuration $c = (name, fullname, Lev, Low, Low, 0.90)$, where *Low* represents the lowercase transformation, yields as links the Cartesian product of all the source and target individuals except (id_2, id_B), resulting in $g(c) = 0.53$. If the threshold in c is changed to 0.65, then $L(c) = \{(id_1, id_A), (id_2, id_D), (id_3, id_C), (id_4, id_D)\}$ and $g(c) = 0.86$. Finally, using 0.50 as threshold, $L(c) = \{(id_1, id_A), (id_3, id_C), (id_4, id_D)\}$ and $g(c) = 1$.

As it can be noted in the previous example, configurations whose generated links contain a large number of unique individuals yield higher singularity scores than other configurations. In general, this is desired when discovering links among RDF datasets; however, users can exploit singularity thresholds to suggest other configurations whose generated links do not contain a large number of unique individuals, but still are promising for link discovery.

4 EVALUATION

We implemented our approach using Apache Jena[2]. In our experiments, we treat all objects as string values, so we used a Java library[3] implementing seven normalized string distances as follows: common subsequence, cosine, Jaccard, Jaro-Winkler, Levenshtein, longest n-grams, and Sorensen-Dice. Furthermore, we used five string transformations as follows: lowercase, remove non-ASCII characters, remove IRI prefix, uppercase, and void (no transformation). We ensure the reproducibility of our results by making our implementation and results publicly available[4].

The rest of this section describes the scenarios we used in our experiments (Section 4.1), comparison with respect to the state of the art and experimental setup (Section 4.2), results and discussion on selecting source and target properties (Section 4.3), and results and discussion on selecting configurations (Section 4.4).

4.1 Scenarios

We evaluated several scenarios from different domains that have been consistently used to evaluate link discovery and entity recognition [23]. Scenarios s_1–s_3 correspond to the Abt-Buy, Amazon-GoogleProducts and DBLP-ACM scenarios from the data matching benchmark[5] devised by Köpcke et al. [22]. Scenario s_4 corresponds to the author recognition task described in the OAEI instance matching challenge of 2015[6], while scenarios s_5–s_7 correspond to the Restaurants, Persons1 and Persons2 scenarios from the OAEI instance matching challenge of 2010[7]. Finally, scenario s_8 corresponds to the sandbox of the SPIMBENCH task from the OAEI instance matching challenge of 2018[8]. Scenarios s_1 and s_2 deal with products, brands, prices and device names; s_3 and s_4 contain papers with their titles, authors and venues; s_5 and s_6 represent people and addresses; s_7 contains restaurants, addresses and cuisine types; and s_8 deals with news items, blog posts and television shows.

Table 1 presents the total number of triples in the source and target datasets ($|D|$), the total number of individuals ($|I|$), and the total number of unique properties that such individuals can be related to ($|P|$). We also present the ground truth provided in terms of total number of individuals ($|I(D_S)|$ and $|I(D_T)|$, respectively), and total number of links ($|L|$). Note that, for all scenarios, ground truths provided do not link all the individuals in the source and/or target datasets. Furthermore, some scenarios contain a larger number of individuals than links.

[2]https://jena.apache.org/
[3]https://github.com/tdebatty/java-string-similarity
[4]https://github.com/crrivero/CHALD
[5]https://dbs.uni-leipzig.de/en/research/projects/object_matching/fever/benchmark_datasets_for_entity_resolution
[6]http://oaei.ontologymatching.org/2015/im/index.html
[7]http://oaei.ontologymatching.org/2010/im/index.html
[8]http://oaei.ontologymatching.org/2018/spimbench.html

Table 1: Summary of the scenarios used in our experiments

	Source			Target			Ground truth		
Id	$\|D\|$	$\|I\|$	$\|P\|$	$\|D\|$	$\|I\|$	$\|P\|$	$\|I(D_S)\|$	$\|I(D_T)\|$	$\|L\|$
s_1	2,162	1,081	2	2,829	1,092	3	1,081	1,092	1,097
s_2	3,974	1,363	3	6,493	3,226	3	1,113	1,291	1,300
s_3	7,848	2,616	3	6,868	2,294	3	2,224	2,224	2,224
s_4	114,425	15,571	8	10,249	1,723	9	854	854	854
s_5	9,000	2,000	14	7,000	1,000	13	500	500	500
s_6	10,800	2,400	14	5,600	800	13	95	400	400
s_7	1,130	339	7	7,520	2,256	7	112	112	112
s_8	9,443	1,126	47	9,411	1,130	67	299	299	299

4.2 Comparison and setup

We compare our approach, which we referred to as CHALD (Configuration Helper for Automated Link Discovery), with respect to SLINT+ [25]. SLINT+ proposes two scores, coverage (*cov*) and discriminability (*dis*), for selecting source and target properties that are combined using the harmonic mean. SLINT+ also proposes to use a number of thresholds to only consider certain source and target properties; however, to compare both approaches, we rank source and target properties based on the combinations of scores proposed by CHALD and SLINT+, and select the top 50% of these properties. In the case of CHALD, we set $\beta = 2$, i.e., we use F_2 to combine universality and uniqueness scores.

Instead of a singularity score that allows to measure promising configurations, SLINT+ proposes a confidence score (*conf*) to only select promising pairs of source and target properties (no distances, transformations or thresholds). A function R is defined over the objects of a given property p that transforms these objects using a number of preestablished rules based on the types of the objects involved, e.g., for IRIs, their domain is omitted and the resulting string is tokenized by separator '/'. The confidence score of a source and a target properties p_s and p_t is computed as follows:

$$conf(p_s, p_t) = \frac{2\,|R(O_S) \cap R(O_T)|}{|R(O_S)| + |R(O_T)|}$$

where O_S (O_T) are all the objects in the source (target) dataset related by property p_s (p_t). For the source and target properties selected in the previous ranking by CHALD and SLINT+, we rank their pairs based on singularity and confidence scores, and select the top 10% of the configurations.

In our experiments, we perform a sweep taking into account all source and target properties in each scenario, transformations (five in total), and distances (seven in total). For each combination of source property p_s, target property p_t, source transformation δ_s, target transformation δ_t and distance ψ, we compute $V = \{v | (i_s, p_s, o_s) \in D_S \wedge (i_t, p_t, o_t) \in D_T \wedge \psi(\delta_s(o_s), \delta_t(o_t)) = v\}$ and use these values as thresholds, i.e., $\theta \in V$. Using each configuration, we compute precision and recall of the links generated by c with respect to the links available as ground truth for each scenario.

4.3 Selection of source and target properties

We first discuss about the selection of source and target properties based on the combinations of scores proposed by CHALD and

Table 2: Rankings of the source properties produced by CHALD and SLINT+ (* means selected)

(a) Scenario s_4

	CHALD				SLINT+			
	Property	F_2	u	q	Property	H	cov	dis
1st	*title**	0.98	0.94	0.98	*author_of**	0.25	0.13	2.63
2nd	*venue**	0.47	0.85	0.42	*title**	0.24	0.13	2.56
3rd	*author_of**	0.22	0.05	1.00	*venue**	0.21	0.12	0.97
4th	*name**	0.22	0.05	0.81	*publisher**	0.15	0.12	0.21
5th	*publisher*	0.12	0.90	0.01	*citations*	0.08	0.10	0.07
6th	*citations*	0.04	0.74	0.03	*year*	0.02	0.12	0.01
7th	*year*	0.01	0.91	0.01	*name*	0.01	0.01	2.30
8th	*rdf:type*	0.00	1.00	0.00	*rdf:type*	0.00	0.27	0.00

(b) Scenario s_7

	CHALD				SLINT+			
	Property	F_2	u	q	Property	H	cov	dis
1st	*has_address**	0.71	0.33	1.00	*name**	0.36	0.2	1.65
2nd	*is_in_city**	0.71	0.33	1.00	*has_address**	0.19	0.10	3.90
3rd	*street**	0.71	0.33	0.99	*is_in_city**	0.19	0.10	3.90
4th	*phone**	0.71	0.33	0.99	*street**	0.19	0.10	3.84
5th	*name*	0.58	0.67	0.56	*phone**	0.19	0.10	3.84
6th	*category*	0.19	0.33	0.17	*year*	0.16	0.10	0.41
7th	*rdf:type*	0.01	1.00	0.01	*rdf:type*	0.03	0.30	0.02

SLINT+, i.e., universality and uniqueness, and coverage and discriminability. There are no differences in the rankings computed by CHALD and SLINT+ for scenarios s_1–s_3; note that all these scenarios are translations of CSV into RDF files and, therefore, they are very uniform, e.g., all instances are related to all properties. Table 2a presents the rankings obtained by CHALD and SLINT+ for the source dataset in s_4. Both approaches select the same first three properties but in different order, i.e., *author_of*, *title*, and *venue*. (For the sake of presentation, we have simplified the IRIs of the properties.) However, CHALD ranks *name* in the fourth position while SLINT+ ranks *publisher* in that position instead of *name*. Even though the universality score of *name* is low, its uniqueness is high and, therefore, our approach prioritizes it with respect to others (note that the *name* property is the penultimate in the ranking output by SLINT+). Configurations that use the *name* property achieve a maximum precision and recall of 0.60 and 1.00, respectively; while configurations that use the *publisher* property achieve a precision and recall of 0 since none of the instances present in the ground truth are related to any objects by the *publisher* property. The selection of *publisher* in favor of *name* by SLINT+ is due to applying the harmonic mean to the coverage and discriminability scores: *name* has a higher discriminability score than *publisher* (2.30 and 0.21, respectively); however, since the coverage score of *name* is lower than the score of *publisher* (0.01 and 0.12, respectively), *publisher* is ranked higher than *name*.

We observe a similar behavior in s_5 and s_6 where SLINT+ ranks first the *name* property while CHALD ranks the same property 10th out of 14 properties and, therefore, it is not selected. Similarly as

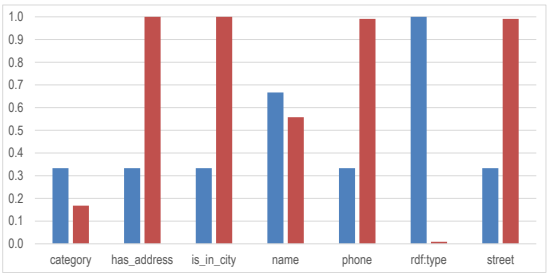


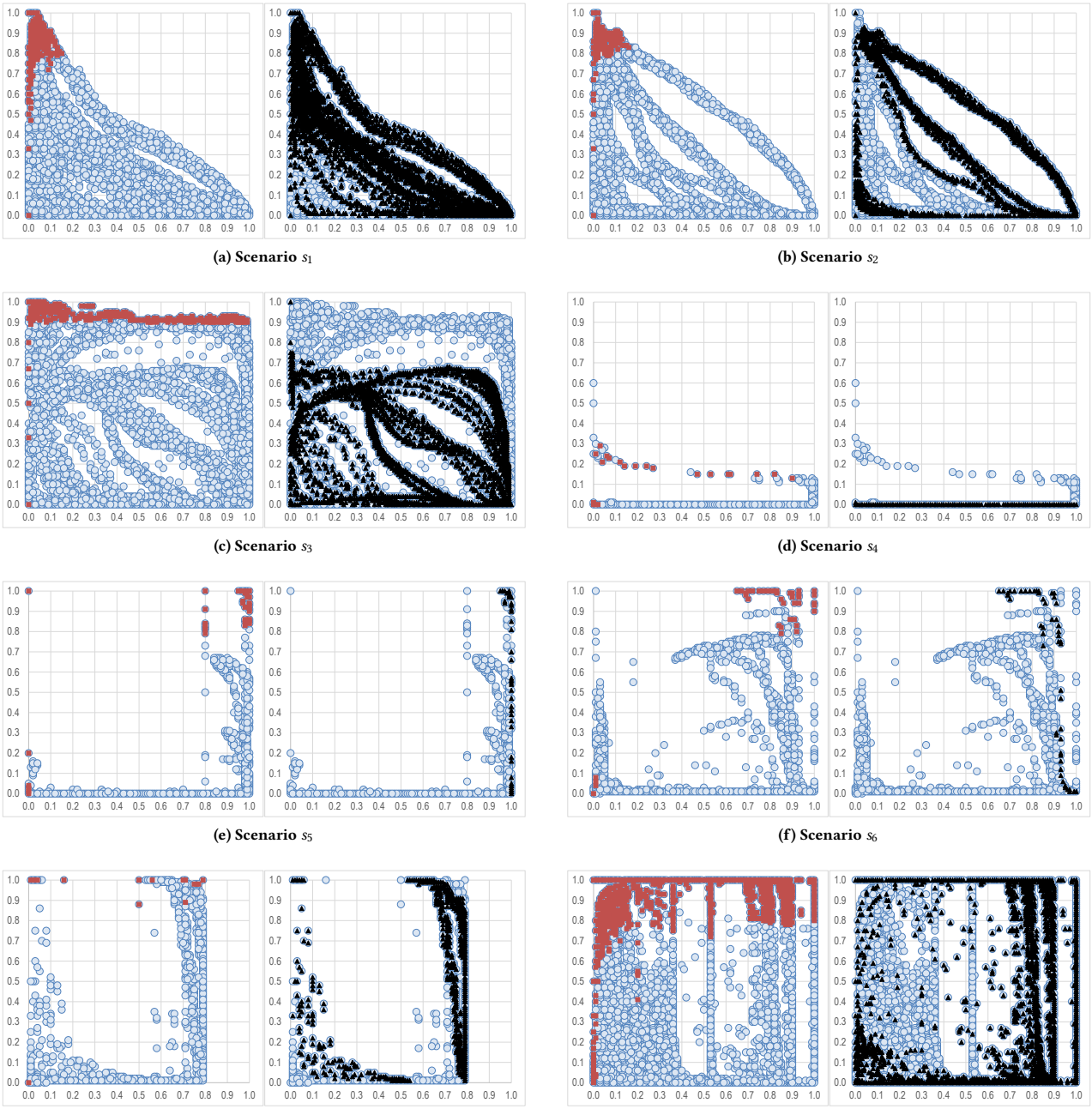
Figure 2: Universality (left bar) and uniqueness (right bar) scores by source property in s_7

before, none of the instances present in the ground truth are related to any objects by *name*, which is used to provide names to streets and states but not persons since, for the latter, the *given_name* and *surname* properties are used. In s_7 (see Table 2b), CHALD does not select the *name* property while SLINT+ selects it in first place; all configurations in this scenario using properties *has_address*, *name* and *phone* achieve precision of 1.00 and recall of 0.79, so the source properties selected by CHALD are also promising. In s_8, we do not observe significant differences between the rankings produced by CHALD and SLINT+.

These results confirm that, in general, uniqueness is preferred with respect to universality since link rules are more likely to produce quality links. Universality and uniqueness scores are relative to the properties available in each dataset at hand; furthermore, another important factor is that the universality score depends on the total number of individuals in a specific dataset, which implies that we also need to take into account the possibility of additional individuals which we are not interested in linking. In addition to rank properties combining universality and uniqueness scores based on F_2, we suggest using plots to help users decide additional properties not selected by the ranking. Figure 2 presents an example of such a plot in which the X axis contains the properties of the datasets, and the Y axis both universality (left bar) and uniqueness (right bar) scores for those properties, respectively. The *category* and *rdf:type* properties are candidates to be discarded since their uniqueness scores are low even though the universality score of *rdf:type* is high. The *name* property can be a promising candidate since its uniqueness score suggests that more than 50% of the individuals have a unique object for this property, which can be helpful when disambiguating certain individuals, i.e., several properties and a certain aggregation function would result in accuracy improvements.

4.4 Selection of configurations

Singularity scores in our approach are defined for both a source and a target datasets of interest. In general, a proper selection would be scores closer to one, which implies that links generated by a given configuration include very few repetitions of the same source and target individuals. Figure 3 shows precision and recall obtained for the configurations kept with respect to all possible configurations using universality, uniqueness and singularity rankings in CHALD, and coverage, discriminability and confidence rankings in SLINT+. High precision and recall values are obtained in all scenarios for CHALD, while SLINT+ fails to obtain these values in s_3, s_4, s_6 and



(a) Scenario s_1

(b) Scenario s_2

(c) Scenario s_3

(d) Scenario s_4

(e) Scenario s_5

(f) Scenario s_6

(g) Scenario s_7

(h) Scenario s_8

Figure 3: Precision (Y axis) vs. recall (X axis) of links generated by all configurations (blue circles), and links restricted by CHALD (left, red squares) and SLINT+ (right, black triangles) in each scenario

s_7. We also observe that CHALD reduces the number of output configurations with respect to SLINT+: on average, the reduction ratio for CHALD is 95% while for SLINT+ is 85%.

We note that, in general, CHALD prioritizes precision with respect to recall, which is generally desired in the context of automated link discovery. Link rules that are very specific tend to have high precision but low recall since, in the extreme case, they only compute a single link. The results of these link rules can then be merged to compose the final set of links. We also note that in all scenarios CHALD does not filter out several configurations with very low precision and recall. Further inspection shows that these configurations contain non-promising pairs of source and target

properties that generate a number of links that contain unique individuals. For instance, in s_5, phone and social security number properties have high universality and uniqueness scores; some configurations that compare phone and social security numbers yield links that contain unique persons but represent only false positive and false negative links, thus precision and recall are (close to) zero.

In certain scenarios, we may have one-to-many and/or many-to-one links involving the same individuals several times. In these cases, configurations may rank lower since their singularity scores are lower than other configurations that only generate links involving unique individuals. When users are aware of these situations, they may need to increase the number of configurations retrieved.

5 RELATED WORK

Link discovery is very related to the problem of entity resolution, and schema and ontology matching [12, 21]. Recent techniques have mainly focused on generating link rules using genetic algorithms [6, 8, 10, 17, 19, 24, 30, 32], which are usually supervised, i.e., they require examples of positive and negative links.

Adaptive link discovery aims to recommend a (partial) configuration for link discovery techniques by analyzing the datasets provided as input [23]. Araújo et al. [3] require the selection of source individuals that have at least a property and a value in common; target individuals are selected until finding a link among them. This approach assumes direct matching in which values are compared without using any distances and/or transformations.

In Freitas et al. [11], for each pair of values in the source and target datasets, a number of distances and thresholds are used to classify such a pair as a positive or a negative link; then, links are further classified based on majority and, if there is no consensus, the user needs to make the final decision. The main drawback of this approach is that a majority of combinations classifying a link as positive or negative does not necessarily entail such a link is indeed positive or negative. Furthermore, all combinations must be evaluated over links in order to classify them.

Nikolov et al. [26] present an approach to, given a source dataset, find relevant target datasets to discover links as well as classes of target individuals that are related to source individuals. As a result, this approach requires the existence of an ontology that must also include hierarchies using super- and sub-classes. The same authors propose using genetic algorithms to find distances and thresholds to be used for link discovery without using input examples, but based on measuring the number of individuals linked [27].

Nguyen et al. [25] propose SLINT+ for link discovery without providing any input examples. SLINT+ first aims to extract source and target properties based on coverage and discriminability. On one hand, coverage measures the number of triples that use a given property with respect to all the triples in the dataset. On the other hand, discriminability combines the entropy of unique values with respect to their frequency over total triples. After property selection, SLINT+ aims to select pairs of promising properties relying on custom transformations based on data types. SLINT+ does not exploit any distances for comparing values. The coverage and discriminability scores are related to our universality and uniqueness scores, respectively. On one hand, our universality score is computed by dividing the number of individuals that is related by a given property by the total number of individuals in a given dataset, while the coverage score takes all triples of the dataset into account. Note that the number of triples may be larger than the number of unique individuals in a dataset. On the other hand, the discriminability score is divided into two terms that compute the percentage of unique values with respect to the total number of triples, and the entropy of the frequency of values. Similarly as before, using the number of unique individuals instead of the total number of triples yields a more accurate percentage. Finally, SLINT+ proposes a confidence score to suggest promising pairs of source and target properties that does not suggest promising configurations including transformations and/or distances.

Hassanzadeh et al. [15] present a generic framework to detect pairs of properties for which the individuals in such datasets are linked. The framework consists of a full suite for link discovery over JSON-like datasets that include a variety of similarity distances and transformations, and additional functions to aggregate, filter and rank discovered pairs of properties. Furthermore, the framework also contains several algorithms to efficiently discover pairs of properties. The authors propose strength and coverage scores measured respectively as the percentage of unique individuals, and total number of individuals for a given pair of source and target properties. The main drawback of these scores is that they require to materialize the links generated for a given configuration, while our universality and uniqueness scores avoid such materialization since they are applied directly over the source and target datasets.

Universality and uniqueness are related to the computation of keys in RDF datasets. A key is a set of properties whose values uniquely identify individuals in a given dataset. Approaches for computing keys mainly focus on a single RDF dataset [1], with some notable exceptions [2, 4, 13, 20, 28, 33]. These approaches mainly consider datasets that follow the unique name assumption, i.e., an individual in a dataset can be linked only to a single individual in another dataset, which may restrict their applicability in practice. Furthermore, they require ontology information since individuals are only considered if they belong to certain classes, or properties in different datasets need to be aligned beforehand since they only consider a single dataset. None of the previous approaches consider distances and transformations but only plain values.

The approach by Atencia et al. [4] evaluates keys based on coverage and discriminability. On one hand, coverage measures the number of individuals that are linked by a key based on the types they belong to (ontology information). On the other hand, discriminability measures how close links generated by a key are to "perfect" one-to-one links, i.e., a source individual is only linked to a target individual. Similarly as the approach by Hassanzadeh et al. [15], links need to be materialized in order to select a suitable pair of source and target properties, while our approach does not require such materialization. Furthermore, discriminability is computed as the minimum number of unique source or target individuals present in a set of links, which does not provide a complete measure since the maximum number of unique individuals are not considered.

6 CONCLUSIONS

This paper presents our approach to select suitable properties in individual datasets to be linked, and then recommend suitable

configurations including a pair of properties, a distance, a pair of transformations and a threshold to link a source and a target datasets. For the individual datasets, we propose universality and uniqueness scores measured as the ratio of individuals related to a given property, and the ratio of unique values related to a given property, respectively. Furthermore, the singularity score measures the entropy of the individuals present in the links generated by a given configuration. Our experiments show that, using universality, uniqueness and singularity scores, we are able to reduce the search space of configurations for link discovery while maintaining high quality configurations based on precision and recall.

Relying on universality, uniqueness and singularity scores may be challenging, so we provide the following guidelines to enable the use of these scores in third-party scenarios: 1) We generally wish properties with high universality scores; however, since this score depends on the total number of individuals, we need to take it into account in the context of other properties in the same dataset. 2) We also wish to keep properties whose uniqueness scores are high since their values are more likely to produce quality links; however, this score does not depend on other individuals present in the dataset. 3) In some datasets, we may wish to keep properties whose universality scores are low but uniqueness scores are high; this is the case in which a property is only related to a small number of individuals whose values are almost unique. We recommend to combine both scores using F_β, $\beta > 1$ to prioritize uniqueness with respect to universality. 4) Selecting top-k positions in rankings based on the combination of universality and uniqueness scores may not retrieve all promising properties in certain cases. In addition to rank properties, we recommend to use plots to visualize scores for all properties in a given dataset. 5) Datasets that contain ambiguity can be detected by universality and uniqueness score plots. We expect such datasets to have a variety of properties whose universality scores are high but uniqueness scores are low. These cases entail that multiple properties need to be aggregated in order to produce quality links, which is responsibility of the automated link discovery technique at hand. 6) We are generally interested in configurations whose singularity scores are high, which entails that mostly unique source and target individuals appear in the links generated. However, this may not be the case in scenarios where we expect one-to-many and/or many-to-one links involving the same individuals several times, which we expect to be known by users in advance. 7) Singularity can be combined with other measures to help users select suitable configurations, such as a range of number of links expected based on source and/or target individuals, or a range of unique source and/or target individuals expected.

ACKNOWLEDGMENTS

This work was supported by the Spanish R&D&I program under grant TIN2016-75394-R.

REFERENCES

[1] Manel Achichi, Zohra Bellahsene, Mohamed Ben Ellefi, and Konstantin Todorov. 2019. Linking and disambiguating entities across heterogeneous RDF graphs. *JWS* 55 (2019), 108–121.
[2] Manel Achichi, Mohamed Ben Ellefi, Danai Symeonidou, and Konstantin Todorov. 2016. Automatic Key Selection for Data Linking. In *EKAW*. 3–18.
[3] Samur Araújo, Duc Thanh Tran, Arjen P. de Vries, and Daniel Schwabe. 2015. SERIMI: Class-Based Matching for Instance Matching Across Heterogeneous Datasets. *TKDE* 27, 5 (2015), 1397–1410.
[4] Manuel Atencia, Jérôme David, and Jérôme Euzenat. 2014. Data interlinking through robust linkkey extraction. In *ECAI*. 15–20.
[5] Christian Bizer, Tom Heath, and Tim Berners-Lee. 2009. Linked Data - The Story So Far. *IJSWIS* 5, 3 (2009), 1–22.
[6] Eduardo N. Borges, Moisés G. de Carvalho, Renata de Matos Galante, Marcos André Gonçalves, and Alberto H. F. Laender. 2011. An unsupervised heuristic-based approach for bibliographic metadata deduplication. *IPM* 47, 5 (2011), 706–718.
[7] Douglas Burdick, Ronald Fagin, Phokion G. Kolaitis, Lucian Popa, and Wang-Chiew Tan. 2019. Expressive power of entity-linking frameworks. *JCSS* 100 (2019), 44–69.
[8] Andrea Cimmino, Carlos R. Rivero, and David Ruiz. 2016. Improving Link Specifications using Context-Aware Information. In *LDOW*.
[9] Richard Cyganiak, David Wood, and Markus Lanthaler. 2014. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. https://www.w3.org/TR/rdf11-concepts/
[10] Moisés G. de Carvalho, Alberto H. F. Laender, Marcos André Gonçalves, and Altigran Soares da Silva. 2012. A Genetic Programming Approach to Record Deduplication. *TKDE* 24, 3 (2012), 399–412.
[11] Junio de Freitas, Gisele L. Pappa, Altigran Soares da Silva, Marcos André Gonçalves, Edleno Silva de Moura, Adriano Veloso, Alberto H. F. Laender, and Moisés G. de Carvalho. 2010. Active Learning Genetic programming for record deduplication. In *CEC*. 1–8.
[12] Jérôme Euzenat and Pavel Shvaiko. 2013. *Ontology Matching, Second Edition*. Springer.
[13] Houssameddine Farah, Danai Symeonidou, and Konstantin Todorov. 2017. KeyRanker: Automatic RDF Key Ranking for Data Linking. In *K-CAP*. 7:1–7:8.
[14] Lise Getoor and Ashwin Machanavajjhala. 2012. Entity Resolution: Theory, Practice & Open Challenges. *PVLDB* 5, 12 (2012), 2018–2019.
[15] Oktie Hassanzadeh, Ken Q. Pu, Soheil Hassas Yeganeh, Renée J. Miller, Lucian Popa, Mauricio A. Hernández, and Howard Ho. 2013. Discovering Linkage Points over Web Data. *PVLDB* 6, 6 (2013), 444–456.
[16] Tom Heath and Christian Bizer. 2011. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool Publishers.
[17] Robert Isele and Christian Bizer. 2012. Learning Expressive Linkage Rules using Genetic Programming. *PVLDB* 5, 11 (2012), 1638–1649.
[18] Robert Isele and Christian Bizer. 2013. Active learning of expressive linkage rules using genetic programming. *JWS* 23 (2013), 2–15.
[19] Robert Isele, Anja Jentzsch, and Christian Bizer. 2012. Active Learning of Expressive Linkage Rules for the Web of Data. In *ICWE*. 411–418.
[20] Anja Jentzsch, Hannes Mühleisen, and Felix Naumann. 2015. Uniqueness, Density, and Keyness: Exploring Class Hierarchies. In *COLD*.
[21] Hanna Köpcke and Erhard Rahm. 2010. Frameworks for entity matching: A comparison. *DKE* 69, 2 (2010), 197–210.
[22] Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. *PVLDB* 3, 1 (2010), 484–493.
[23] Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. 2017. A survey of current Link Discovery frameworks. *SemWeb* 8, 3 (2017), 419–436.
[24] Axel-Cyrille Ngonga Ngomo and Klaus Lyko. 2012. EAGLE: Efficient Active Learning of Link Specifications Using Genetic Programming. In *ESWC*. 149–163.
[25] Khai Nguyen, Ryutaro Ichise, and Bac Le. 2012. SLINT: A schema-independent linked data interlinking system. In *OM*.
[26] Andriy Nikolov, Mathieu d'Aquin, and Enrico Motta. 2011. What Should I Link to? Identifying Relevant Sources and Classes for Data Linking. In *JIST*. 284–299.
[27] Andriy Nikolov, Mathieu d'Aquin, and Enrico Motta. 2012. Unsupervised Learning of Link Discovery Configuration. In *ESWC*. 119–133.
[28] Nathalie Pernelle, Fatiha Saïs, and Danai Symeonidou. 2013. An automatic key discovery approach for data linking. *JWS* 23 (2013), 16–30.
[29] Thomas Rebele, Fabian M. Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. YAGO: A Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames. In *ISWC*. 177–185.
[30] Amit Singh and Aditi Sharan. 2017. Adaptive genetic programming based linkage rule miner for entity linking in Semantic Web. In *ICCCA*. 373–378.
[31] Tommaso Soru and Axel-Cyrille Ngonga Ngomo. 2014. A comparison of supervised learning classifiers for link discovery. In *SEMANTICS*. 41–44.
[32] Chenchen Sun, Derong Shen, Yue Kou, Tiezheng Nie, and Ge Yu. 2017. A genetic algorithm based entity resolution approach with active learning. *FCSC* 11, 1 (2017), 147–159.
[33] Danai Symeonidou, Vincent Armant, Nathalie Pernelle, and Fatiha Saïs. 2014. SAKey: Scalable Almost Key Discovery in RDF Data. In *ISWC*. 33–49.