# Filtering Inaccurate Entity Co-references on the Linked Open Data

John Cuzzola[1(✉)], Ebrahim Bagheri[1], and Jelena Jovanovic[2]

[1] Ryerson University, Toronto, ON, Canada
{jcuzzola, bagheri}@ryerson.ca
[2] University of Belgrade, Belgrade, Serbia
jeljov@gmail.com

**Abstract.** The Linked Open Data (LOD) initiative relies heavily on the interconnections between different open RDF datasets where RDF links are used to connect resources. There has already been substantial research on identifying identity links between resources from different datasets, a process that is often referred to as co-reference resolution. These techniques often rely on probabilistic models or inference mechanisms to detect identity relations. However, recent studies have shown considerable inaccuracies in the LOD datasets that pertain to identity relations, e.g., owl:sameAs relations. In this paper, we propose a technique that evaluates existing identity links between LOD resources and identifies potentially erroneous links. Our work relies on the position and relevance of each resource with regards to the associated DBpedia categories modeled through two probabilistic category distribution and selection functions. Our experimental results show that our work is able to semantically distinguish inaccurate identity links even in cases when high syntactical similarity is observed between two resources.

**Keywords:** Resource co-reference · Linked open data · Semantic web · Semantic disambiguation · Identity links

## 1 Introduction

Linked Data relies on establishing relationships between resources, such as the equivalence (identity) relationship, that span across different ontologies and datasets [1]. Identity links such as those based on the owl:sameAs property are commonly used for establishing equivalence relationships by asserting that two different URIs refer to the same resource [10]. The importance of identity links is primarily due to the significant role they play in interconnecting different datasets of the Linked Open Data (LOD) cloud. Consequently, mistakes in these linkages may result in erroneous assertions by applications that automatically traverse these identity links [4]. In practice, identity links such as the owl:sameAs links tend to exhibit two main problems:

(P1) First, there are obvious errors where resources represented by URIs are clearly neither similar nor related. For example, based on existing information on the LOD cloud, *dbpedia:Dog* and the derogatory *dbpedia:Bitch(insult)* are considered to be the same. These kinds of mistakes atypical for humans are most likely made by an

Q. Chen et al. (Eds.): DEXA 2015, Part I, LNCS 9261, pp. 128–143, 2015.
DOI: 10.1007/978-3-319-22849-5_10

automated inference machine that failed to properly disambiguate the underlying semantics of the URIs to be matched. In addition, it is also not unlikely that human error in judgment could play some role in such erroneous identity links.

(P2) Second, the determination of co-reference resolution can at times be subjective and dependent on a specific application case. For example, *dbpedia:Evil*, *dbpedia:Morality*, and *dbpedia:Crime* have already been linked together through identity links. However, although these three resources are strongly related, they are not equivalent in that one can have questionable morality and not be evil; likewise, not every criminal act is morally wrong. From a strict interpretation of the owl:sameAs property, no pair of these three resources is identical.

There has already been novel and interesting work that focuses on co-reference resolution with specific attention to URIs (i.e., resources) of the LOD cloud (e.g., [3, 8, 9]). The proposed techniques attempt to identify any potential pair of URIs that can or need to be related to each other through identity links [6]. However, in this paper, we focus on a different aspect of this problem: our focus is on improving the quality of existing identity links that are already part of the LOD cloud. In other words, our objective is to ensure that an existing identity link collection contains reliable co-reference resolutions. To this end, this paper makes the following contributions:

- A novel algorithm that can identify potential mismatches between the URIs that are linked via identity links, and hence provide the means to filter out such mismatches in order to resolve (P1).
- The calculation of an upper-bound semantic score that can be used as a measure of semantic similarity of resources connected via identity links. This way, the algorithm points to cases where identity links show some degree of subjectivity and hence allows for the identification of cases represented in (P2).

## 2 Background

An increasing number of organizations are publishing their data on the Web as Linked (Open) Data thus continuously extending the Web of Data with new datasets. However, while the number of datasets keeps growing, linking between those datasets keeps lagging behind, thus leading to a considerable discrepancy between intra- and inter-dataset linking. In other words, while data tend to be well connected within individual datasets, linking between datasets is still not at the level required for a true Web of Linked Data. Based on a recent analysis of the crawlable subset of the LOD cloud (April 2014), Schmachtenberg et al. [13] reported that 56.11 % of the crawled datasets are connected to at least one other dataset, while the rest of the datasets are either only the targets of RDF links or are completely isolated. In addition, only a small number of datasets is highly linked, while the majority of them are only sparsely linked.

Aiming to improve the connectivity of the Web of Data, the Semantic Web research community has proposed several methods for automated or semi-automated linking of resources on the Web of Data, especially resources originating from disparate datasets [5, 14]. While these proposals promise to make a notable contribution to the realization of the Web of Linked Data, they also tend to generate erroneous links that over time can negatively impact the overall quality of the Web of Data [7]. Such quality issues are of particular concern since unlike the Web of Documents where humans can examine the meaning and usefulness of a link, on the Web of Data such an examination is left to the software agents harvesting and making use of the data.

While links of diverse semantics can be used to connect resources from disparate datasets, in this paper we focus exclusively on identity links connecting two resources that are considered either identical or closely related [15]. Such links are often referred to as sameAs links even though they do not necessarily originate from the owl:sameAs property, but also from other properties that have been used for establishing identity links on the Web of Data (e.g., skos:closeMatch, skos:exactMatch). Still, owl:sameAs links tend to be dominant among identity links. According to the aforementioned study by Schmachtenberg et al. [13], owl:sameAs is among the top three most used linking predicates in 7 out of 8 topical categories of the LOD cloud.

## 3 Proposed Approach

The main objective of our work is to flag and possibly remove questionable sameAs links from the LOD cloud. To achieve this objective, we define a two step process: (1) calculate a similarity score between the URIs involved in a given sameAs link, which would represent the likelihood that the URIs are in fact referring to the same resource; and (2) employ this similarity score to determine whether the sameAs link is valid and reliable or needs to be flagged for removal. Therefore questionable sameAs links can be flagged (and possibly filtered) if they do not meet some minimum threshold value; this value is chosen based on empirical studies reported in Sect. 6 (Fig. 2).

We refer to our proposed method as Semantic Co-reference Inaccuracy Detection (SCID). We begin by introducing the methods and formulas that form the foundation of our algorithm (Sect. 4), and then introduce the algorithm itself in Sect. 5.

## 4 The Components of SCID

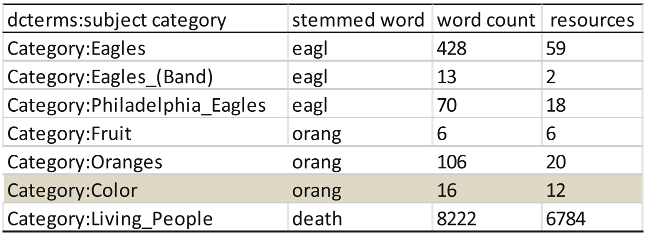
Our strategy for detecting inaccurate identity links can be summarized as follows. Assume we construct a baseline vector $v_x$ that is known to semantically represent resource $x$. We create $v_x$ based on DBpedia categories that relate to $x$ (positive categories) as well as categories that do not relate to $x$ (negative categories). We denote this combined set of categories as $\mathrm{S_x}$. Given a resource $y$ of unknown similarity to $x$, we can construct $v_y$ using the same categories $\mathrm{S_x}$. We can now compare similarity of two vectors using measures of similarity such as Pearson correlation coefficient to judge if $v_y$ is “same as” $v_x$. Our method relies on three key components:

1. *Frequency count statistics*. We require a listing of the most frequently occurring words within each category of DBpedia’s 995,911 subject categories[1]. In Sect. 4.1 we explain the process of computing the required data.

[1] DBpedia version 3.9.

2. *A category distribution function*. This is the core function of our method. Given some input text obtained from resources x and y; and an arbitrary subset of DBpedia subject categories S, the function produces vectors $v_x$ and $v_y$ that can be compared to identify erroneous identity links. This is explained in Sect. 4.2.
3. *A category selection function*. The category distribution function requires a subset of DBpedia subject categories as input. This function, explained in Sect. 4.3, provides a mechanism for selecting this subset of categories.

**Table 1.** A sample of word and resource frequency counts within specific DBpedia categories (category:Color highlighted).



| dcterms:subject category | stemmed word | word count | resources |
|---|---|---|---|
| Category:Eagles | eagl | 428 | 59 |
| Category:Eagles_(Band) | eagl | 13 | 2 |
| Category:Philadelphia_Eagles | eagl | 70 | 18 |
| Category:Fruit | orang | 6 | 6 |
| Category:Oranges | orang | 106 | 20 |
| Category:Color | orang | 16 | 12 |
| Category:Living_People | death | 8222 | 6784 |

### 4.1 Frequency Count Statistics

***Problem Outline:*** In order to train a model for inaccuracy detection, we require frequency counts over DBpedia categories for use in the category distribution function of SCID.

SCID centers around the dcterms:subject property of the DBpedia resources. This property provides 900,000+ subject-matter categories for approximately 11 million DBpedia resources. SCID is trained around five summary statistics (*features*) that rely on resource categories; these are described in detail in Sect. 4.2.

Table 1 is an example of word frequency counts for resources that belong to DBpedia categories {*Eagles, Eagles_(Band), Philadelphia_Eagles, Fruit, Orange, Color, Living_People*}. In this table, the row of the stemmed word "orang" and the category Color is highlighted; it indicates that there are 12 resources that refer to the category Color and use the word orange. Furthermore, the stem "orang" appears 16 times within this set of 12 resources; thus the average frequency of orange in this set is 16/12 or 1.3. These types of statistics are formally defined in Sect. 4.2.

### 4.2 The Category Distribution Function

***Problem Outline:*** Given an input text and a set of DBpedia subject-matter categories, we require a normalized vector to represent the relevancy of each category to the input text.

The core of our method is the category distribution function $v = \rho(t,S)$ where $t$ is an input text to be processed, and S is a specified subset from the DBpedia subject categories. The output is a vector *v* representing a proportionate mixture of each of the categories of S as they relate to the input text *t*. Table 2 illustrates the function with three DBPedia categories: *Eagles*, *Eagles_(band), and Philadelphia_Eagles* on three input texts. This example demonstrates that our method can associate different usages of a word with its appropriate category.

**Table 2.** The output of the category distribution function ρ(t,S) on S = [Eagles, Eagles_(band), Philadelphia_Eagles] for three sample inputs (t).

| | **Natural Language Input Text (t)** | | |
|---|---|---|---|
| **Category Subset (S)** | Sproles, who was acquired by the Eagles in the off- season, led the league with 2,696 all- purpose yards in 2011, but his rushing.. | The Eagles were formed in 1971 by guitarist / singer Glenn Frey, With an eye towards his future band, he approached Henley to be her drummer. | The eagle is one the largest and most powerful birds of prey. Soaring high above the earth, spying its prey with its keen eyes. |
| Eagles | 0.086 | 0.214 | 0.72 |
| Eagles_Band | 0.207 | 0.623 | 0.13 |
| Philadelphia_Eagles | 0.706 | 0.162 | 0.148 |
| ***P(t,S)*** | v=[0.086,0.207,0.706] | v=[0.214,0.623,0.162] | v=[0.72,0.13,0.148] |

A more interesting example is given in Table 3 with chosen categories *Fruit*, *Oranges*, and *Color,* which are not as disjoint as those in Table 2. Specifically, the Fruit category is defined as a category broader than Oranges (via the skos:broader property). Moreover, the mention of color in the input text adds a disambiguation challenge w.r.t. the Oranges and Color categories. Further adding to the complexity is that the input text refers to a specific color (dark pink), but that color is not orange; the term orange in this context refers to the Fruit category. The output of the distribution function is a proportional mixture of the three relevant categories.

**Table 3.** ρ(t,S) output when categories [Fruit, Oranges, Color] are related/ambiguous w.r.t term "orange" and the input text refers to all three categories.

| | **Natural Language Input Text (t)** |
|---|---|
| **Category Subset (S)** | Cara Cara , a type of navel orange, are also available during the winter months. They are like the familiar Washington navels, but the fruit's interior is dark pink. |
| Fruit | 0.274 |
| Oranges | 0.504 |
| Color | 0.220 |
| ***P(t,S)*** | v = [0.274,0.504,0.220] |

We now detail the sequence of calculations required for ρ(t,S). Formally, let T be a set of stemmed words from input text *t*; for simplicity of expression, in the following we refer to elements of the set T as words (instead of stemmed words). Let S be the set of pre-chosen categories. Let $W_{x,y}$ be the frequency count of a stemmed word *x* in category *y*.

***Feature Formalization*** $f^1$: Define $f^1_{j,k}$ as the frequency of a specific word *j* in the category *k* to the count of all the words of the input text *t* within the category *k*.

$$f^1_{j,k} = \frac{W_{j,k}}{\sum_{x \in T} W_{x,k}} \text{where} j \in T, k \in S \tag{1}$$

Conceptually, this feature is a measure of the significance of a word relative to a specific category.

***Feature Formalization*** $f^2$: Let $f^2_{j,k}$ be the frequency of specific word *j* in the category *k* to the total frequency count across all the words of the input text within all the categories of *S*.

$$f^2_{j,k} = \frac{W_{j,k}}{\sum_{x \in T} \sum_{y \in S} W_{x,y}} \text{where} j \in T, k \in S \tag{2}$$

Conceptually, this feature is a measure of the importance of a word (e.g., orange) relative to all selected categories (Fruit, Oranges, Color) combined.

***Feature Formalization*** $f^3$: Let $D_{j,k}$ be the number of DBpedia resources that belong to the category *k* and contain the word *j*. We define $f^3_{j,k}$ as the ratio of DBpedia resources that contain word *j* and belong to the category *k* to the number of resources that contain any of the words from the input text that belongs to category *k*. Formally:

$$f^3_{j,k} = \frac{D_{j,k}}{\sum_{x \in T} D_{x,k}} \text{where} j \in T, k \in S \tag{3}$$

This feature is similar to $f^1$ except that instead of counting the frequency of every word occurrence, $f^3$ counts the number of unique resources containing that word.

***Feature Formalization*** $f^4$: Conceptually similar to $f^2$, we define $f^4_{j,k}$ as the ratio of the number of DBpedia resources containing specific word *j* from category *k* across the number of all the resources containing any word from the input text *t* that belongs to any category from the chosen set S. Like $f^3$, this feature deals with the number of distinct resources that contain the word rather than the word frequency count.

$$f^4_{j,k} = \frac{D_{j,k}}{\sum_{x \in T} \sum_{y \in S} D_{x,y}} \text{where} j \in T, k \in S \tag{4}$$

***Feature Formalization*** $f^5$: This last feature is defined as the ratio of the frequency of the word *j* within the category *k* to the total number of resources that belong to *k* and contain *j*. Formally, it is a measure of the average word frequency per resource.

$$f_{j,k}^{5} = \alpha \frac{W_{j,k}}{D_{j,k}} \text{where} j \in T, k \in S \tag{5}$$

and $\alpha$ is a normalizing constant such that $\sum_{y \in S} f_{j,y}^{5} = 1$.

***Word Importance*** $R_{j,k}$: We can now combine all the features to compute the importance of the word $j$ relative to the category $k$. Let $\mathrm{U_k}$ be the total number of resources that belong to the category $k$ globally within the DBpedia knowledge base. Let $\mathrm{O_j}$ be the frequency count of stemmed word $j$ within the input text $t$. The importance of the word $j$ relative to the category $k$ becomes:

$$R_{j,k} = O_j \times \frac{D_{j,k}}{U_k} \times \frac{\sum_{i=1}^{5} f_{j,k}^{i}}{5} \text{where} j \in T, k \in S \tag{6}$$

***The Category Distribution Function*** $\rho(t, S)$: We sum the importance of all the words per chosen category to construct the vector $v$ of the category distribution function and normalize:

$$v = \rho(t, S) = \alpha \left[ \sum_{x \in T} R_{x,k_1}, \ldots, \sum_{x \in T} R_{x,k_n} \right] \tag{7}$$

Where $j \in T, k \in S$ and $\alpha$ is a normalizing constant such that $\sum_{y=1}^{n} \sum_{x \in T} \alpha R_{x,k_y} = 1$.

The final artifact is Eq. 7 that produces the output seen in Tables 2 and 3. We have made available an online implementation of the category distribution function for those interested in further experimentation[2].

### 4.3 The Category Selection Function

***Problem Outline:*** We require a well-defined method for selecting a subset of categories from amongst the 995,911 DBpedia categories to be used as the input set S in $\rho(t, S)$.

The category distribution function (Sect. 4.2) requires as its input a set of subject matter categories for consideration. For example, in Table 3 the input categories were explicitly given as [Fruit, Oranges, Color]. A focused selection of categories is required because the evaluation of all available DBpedia categories is not feasible for computational and practical reasons. Consequently, a strategy for selecting a suitable subset of categories is the focus of this section. Ideally, the selection should include both categories that are related to each other such as Fruit and Oranges in the case of Table 3, as well as disjoint categories such as Eagles, Eagles_(band), and Philadelphia_Eagles in the case of Table 2. We take advantage of DBpedia disambiguation resources to this end. Such resources often encompass homonyms that require examination of the context to differentiate between ambiguous resources. Suppose we wish to find the categories S that will be used to validate the <sameAs> identity links for $x$ = *dbpedia:*

[2] http://ls3.rnet.ryerson.ca/predicatefinder/category/.

*Red*. We see that x belongs to *uri = dbpedia:Red_(disambiguation)* alongside 116 other resources some of which are disambiguation resources themselves. Specifically, *x* and *uri* satisfy the constraint:

$$\{\text{?uri dbpedia-owl : wikiPageDisambiguates ?x}\} \tag{8}$$

Now, let $\mathrm{N_{uri}}$ be the set of resources linked from a DBpedia disambiguation URI. Formally, if $x \in N_{uri}$ then $x$ satisfies Eq. 8. Next let C(x) be a function returning the set of categories for the resource *x*. Namely, $y \in C(x)$ when {?x dcterms:subject ?y}. Lastly, we define $\mathrm{S_{uri}}$ as the union of all subject categories for the resources that a disambiguation resource refers to and apply it recursively for resources within to dereference all resources to their respective categories.

$$S_{uri} = \begin{cases} \bigcup_{x \in N_{uri}} C(x), & \text{if x is a non-disambiguation URI} \\ S_x, & \text{otherwise} \end{cases} \tag{9}$$

Suppose no disambiguation resource *uri* exists for *x* (i.e.: no *uri* satisfies Eq. 8). In this circumstance, we create a temporary disambiguation resource $uri_{temp}$ that references the single resource *x* so that Eq. 9 can still be applied.

# 5 The SCID Filter

In this section we apply frequency count statistics, the category distribution function, and the category selection method of Sect. 4 to construct two algorithms to filter out inaccurate identity links. Algorithm 1 produces disambiguation baseline vectors used to train our model. Once trained, Algorithm 2 details how the model is used to test the accuracy of candidate identity URI pairs.

## 5.1 Algorithm 1 – Constructing Disambiguation Vectors

***Problem Outline:*** We require a method to construct a baseline vector $v_{x,S_x}$ to disambiguate an ambiguous resource *x* (e.g., Eagles as: team, a band, or bird) against DBpedia subject categories $(S_x)$ chosen by Eq. 9.

To construct these vectors we use the category distribution function (Sect. 4.2) and define $v_{x,S_x} = \rho(\gamma_x, S_x)$ as a disambiguation baseline vector for resource *x* where $\gamma_x$ are the most frequently occurring words within the subject categories of resource *x*. Algorithm 1 outlines how these words $\gamma_x$ are found and how $v_{x,S_x}$ is computed.

Consider the resource dbpedia:Red with subject categories C(dbpedia: Red) = dbpedia:{Color, Optical_spectrum, Shades_of_red, Web_colors}. In Algorithm 1, we begin with this resource's stemmed words (line 1). In line 2, we collect all the resources that are also associated with any of the C(dbpedia:Red) categories and include them in our frequency counts. We discard those categories that contain more than 1000 resources because they are overly broad and are not representative of the target resource dbpedia:Red. In line 3, we keep the most frequently occurring words from this collection (75th percentile) and z-score normalize the frequency counts (line 5). We perform this normalization to balance the influence of a more frequent category (e.g., Color) against a less used category (e.g., Web_colors). Finally we compute the disambiguation baseline vector $v$ (line 6).

```
Input (x): A resource URI (x) and a set of subject categories ($S_x$) from equation 9.
Output ($v_{x,S_x}$): A disambiguation baseline vector for resource x.
Algorithm:
Define: Let H(x) be a set of stemmed words appearing in resource x. Let C(x) be the set of categories associated with the resource x. Namely, $y \in C(x)$ iff (?x dcterms:subject ?y). Let $C^{-1}(y)$ be the set of resources that have category y. Namely, if $x \in C^{-1}(y)$ then $y \in C(x)$.
    1. Initialize $\beta \leftarrow H(x)$
    2. For each $y \in C(x), K \leftarrow C^{-1}(y), if\ |K| \leq 1000\ then\ \forall k \in K, \beta \leftarrow \beta + H(k)$
Define: Let $I_{75}$ ($\beta$) be the top 25% (75th percentile) of the most frequently occurring words of set $\beta$. Let $Z_{75}(\beta, h)$ be the z-score normalized frequency count of word $h$ from set $\beta$.
    3. $\beta \leftarrow I_{75}(\beta)$
    4. Initialize $\gamma_x \leftarrow \emptyset$
    5.For each word of $\beta$, append to $\gamma_x$ the word $Z_{75}$ times.
            Namely, for each $\omega \in \beta, \gamma_x \leftarrow \gamma_x + \omega \times Z_{75}(\beta, \omega)$.
    6. Compute disambiguation baseline vector $v_{x,S_x} = \rho(\gamma_x, S_x)$.
```

**Algorithm 1.** Computation of disambiguation baseline vector $v_{x,S_x}$.

The categories of C(dbpedia:Red) form a subset of the category candidate list (S) for the dbpedia:Red_(disambiguation) resource (Sect. 4.3). We note C to contain the target categories with respect to the resource dbpedia:Red while S\C are the noisy categories of the disambiguation vector. The noisy categories are used to counterbalance the frequently occurring words of the target categories with the occurrence of the words in the non-related category set. This provides positive and negative category usage examples and aids in disambiguation. In the next section, we show how the disambiguation baseline vector $v_{x,S_x}$ can be used to find identity link errors.

## 5.2 Algorithm 2 – Detection of Inaccurate Identity Links

***Problem Outline:*** We require a method to utilize the disambiguation baseline vectors $v$ for identifying likely errors in a collection of identity links.

Algorithm 2 details our method for filtering out likely errors within a collection of identity links. In line 1, using Algorithm 1 (Sect. 5.1), we compute a base vector $v_{x,S_x}$ for a resource (x) we wish to validate. We then collect a set of all identity links for the resource, say M(x), using a database of identity links (e.g., www.sameas.org); then, for each identity link $m \in$M(x), we calculate vector $\rho(y_m, S_x)$ where $y_m$ is some descriptive text of $m$ (e.g., rdfs:comment) (line 2). The text of $y_m$ is chosen based on the origin of the candidate URI. Specifically, we use the *rdfs:comment* property when the candidate

URI is from DBpedia or OpenCyc[3]. We use *ns.common.topic.description* property when the URI is from Freebase[4], and *wn20schema:gloss* when it originates from WordNet[5]. The output of line 2 can be compared with the disambiguation baseline vector $v$ of line 1 using some similarity measure to produce a *semantic relatedness score*. We create a *disambiguation ratio* by normalizing semantic relatedness scores using the largest seen score (line 3). Finally, we flag those identity links (URIs) that do not meet a threshold value set for the disambiguation ratio (line 4).

Table 4 shows the results of this algorithm applied to the *dbpedia:Port* resource. The table includes the semantic relatedness score as a measure of the semantic similarity between the candidate URI and the baseline vector (for dbpedia:Port); it is computed using Pearson correlation coefficient as the similarity measure (ß). If the normalized ß (disambiguation ratio) is less than the given threshold (i.e.: ß < $\delta$) then the resource should be flagged as a possible inaccuracy. The isSame classification (Table 4, col 1), determined by a human oracle, indicates whether the candidate URI is truly the same or semantically similar.

```
Input (x, $\delta$): A URI (x) for identity validation and a minimum disambiguation threshold value($\delta$).
Output (Q): A set (Q) of identity links (URIs) for resource x that meet the minimum disambiguation threshold value ($\delta$).
Algorithm:
    1.Compute disambiguation baseline vector $v_{x,S_x}$ using Algorithm 1.
Define: Let $y_x$ be a natural language text description for resource x. Let $Coeff(v_1, v_2)$ be the similarity measure between two vectors $v_1$ and $v_2$. Let M(x) be the set of identity links (URIs) for resource x from an identity links collection (e.g.: sameAs,org).
    2. For each $m \in M(x)$ calculate set $Q$=[$q_m$, ...] where vector $q_m = \rho(y_m, S_x)$.
    3. For each $q_m \in Q$ calculate $\tau_m = Coeff(q_m, v_{x,S_x})$. Let $J_{max}$ be the maximum $\tau_m$ encountered.
    4. Discard $q_m \in Q\ if\ \frac{\tau_m}{J_{max}} < \delta$
```

**Algorithm 2**. A method for discovering inaccurate identity links.

Notice in Table 4 what appears to be an error in which the second entry of dbpedia: Port obtains a disambiguation score less than itself. This result is consistent with our algorithm in that strictly speaking our method is not comparing dbpedia:port with all other sameAs candidates. Instead, *all sameAs candidates of dbpedia:port is compared to the shared categories of dbpedia:port_(disambiguation) that dbpedia:port* belongs to ($S_x$ in Eq. 9) thus forming a cluster of identity links within the radius of the disambiguation threshold value $\delta$. Conceptually, $v_{x,S_x}$ is the center of a cluster while $\delta$ is the "distance" from this center to the outermost boundary of our solution space. Those

[3] http://sw.opencyc.org/.

[4] https://www.freebase.com/.

[5] http://wordnet.princeton.edu/.

identity links within this boundary are members of the cluster while those outside this area are considered anomalies. Figure 1 illustrates.

**Table 4.** Algorithm 2 applied to the dbpedia:Port sameAs candidates

| isSame [y/n] | SAMEAS URI CANDIDATES | Semantic Relatedness | Disambiguation Ratio | Natural Language Source |
|---|---|---|---|---|
| y | dbpedia.org:List_of_Panamax_ports | 0.633 | 1 | rdfs:comment |
| y | dbpedia.org:Port | 0.591 | 0.933649289 | rdfs:comment |
| y | www.w3.org:synset-seaport-noun-1 | 0.565 | 0.89257504 | wn20schema:gloss |
| y | rdf.freebase.com/ns/en.port | 0.539 | 0.85150079 | ns:common.topic.description |
| y | rdf.freebase.com:..a8561 | 0.539 | 0.85150079 | ns:common.topic.description |
| y | sw.opencyc.org:Seaport | 0.478 | 0.755134281 | rdfs:comment |
| y | dbpedia.org:River_port | 0.385 | 0.60821485 | rdfs:comment |
| n | dbpedia.org:Bad_conduct | 0.168 | 0.265402844 | rdfs:comment |
| n | rdf.freebase.com:m.02ss4j | 0.168 | 0.265402844 | ns:common.topic.description |
| n | rdf.freebase.com:en.military_discharge | 0.168 | 0.265402844 | ns:common.topic.description |
| n | dbpedia.org:IVDP | -0.048 | -0.075829384 | rdfs:comment |
| n | rdf.freebase.com:m.0143h2 | -0.062 | -0.097946288 | ns:common.topic.description |
| n | rdf.freebase.com:en.port_wine | -0.062 | -0.097946288 | ns:common.topic.description |
| n | rdf.freebase.com:..120de2 | -0.062 | -0.097946288 | ns:common.topic.description |

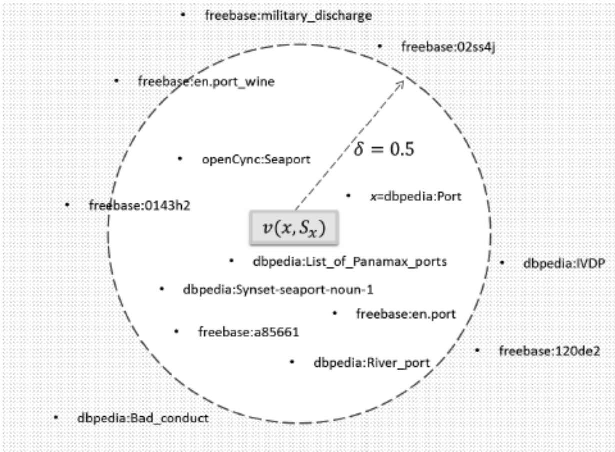


**Fig. 1.** Conceptual diagram for identity links clustered around vector $v$ constructed using the shared categories of *dbpedia:Port_(disambiguation)* resource within threshold $\delta = 0.5$.

## 6 Experimental Evaluation

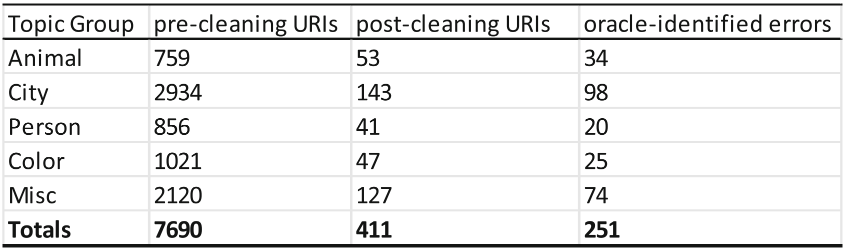
This section presents our experimental evaluation of the SCID approach on a sameAs dataset retrieved from the SameAs.org service[6]. The dataset consists of resources from five groups: Animal, City, Person, Color, and Miscellaneous (other). We decided on

[6] Data retrieved in October 2014 from http://www.sameas.org.

segmenting our dataset into these topical groups to be able to verify whether the performance of SCID is dependent on a specific domain or it performs the same across different non-overlapping domains. We also wanted to discover if there is a general-purpose disambiguation threshold ($\delta$) that would be effective regardless of the topic group. Furthermore, it is important to point out that the scoping of the dataset to specific topics is a common practice within previous studies that typically focused on a narrow domain such as restaurants or people (e.g., [11, 12]). We included the miscellaneous group to further broaden our experiments.

We collected a sameAs dataset of 7,690 candidate URIs for validation (Table 5). First, we performed some necessary data cleansing on these candidates. After this pre-processing procedure, a total of 411 URIs primarily from DBpedia, Freebase, OpenCyc, and WordNet sources remained. A human oracle then identified 251 incorrect URIs (61 % errors) from these 411.

**Table 5.** Experimental dataset showing for each topic group: total candidate URIs (pre/post cleaning) and the number of oracle-identified errors.



| Topic Group | pre-cleaning URIs | post-cleaning URIs | oracle-identified errors |
|---|---|---|---|
| Animal | 759 | 53 | 34 |
| City | 2934 | 143 | 98 |
| Person | 856 | 41 | 20 |
| Color | 1021 | 47 | 25 |
| Misc | 2120 | 127 | 74 |
| **Totals** | **7690** | **411** | **251** |

The data cleansing process includes removing duplicate entries that are aliases or URI redirects for the same resource (such as those identified with the dbpedia-owl: wikiPageRedirects property). Broken-link (non-resolvable) URIs that are no longer accessible are also discarded. Furthermore, because our method relies on a natural language text description of the candidate entry, we discarded those candidates that did not have a well-defined descriptive property namely: rdfs:comment, ns.common.topic. description, or wn20schema:gloss. We ignored duplicates in cases when a candidate URI shared the same descriptive property attribute and same property value with another URI already included in the dataset. A common example is the DBpedia Lite knowledge base that often shares the same rdfs:comment predicate and value with its larger counterpart DBpedia. To illustrate the importance of this purging step, consider the concept *DBpedia:Jesus* that, according to the sameAs.org database, returns 16,889 coreferents of which 3,357 are broken/non-resolvable (20 %), 9,891 are duplicated via aliases/redirects (73 %), and only 3,641 are unique (27 %). Consequently, the large discrepancy between pre and post-cleansing of Table 5 is attributed to aliases and broken links.

Once cleansed, we computed a starting (baseline) F-score value for each of the considered resource groups (Table 5) as an initial measure of the dataset quality. For example, in the City group, the cleaned set of 143 URIs contained 98 errors giving a precision of 0.31. Since this was our starting set, we assumed a recall of 1.0, thus giving a starting F-score of 0.479. We then applied SCID to the cleaned set of candidate URIs at thresholds intervals from 0 to 0.9, and compared the new F-scores against the original baseline. The results of this experiment are summarized in Table 6. The table also provides an average of the five groups for each threshold value as well as a combined F-score tally in which all 411 candidate URIs are evaluated together (non-grouped). We can see a significant improvement from the original non-filtered F-scores for all groups, including the combined group, using any of the disambiguation thresholds with the exception of $\delta \geq$ 0.9. Empirical results indicate that a threshold of 0.5 to 0.6 gives the best results with an average F-score of 0.84.

**Table 6.** F-scores for the five resource groups at varying threshold values ($\boldsymbol{\delta}$), including average score for each group and combined score for all candidate URIs.

| | Original | δ≥ | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Animals | 0.528 | | 0.765 | 0.842 | **0.927** | 0.905 | 0.884 | 0.884 | 0.884 | 0.844 | 0.792 | 0.633 |
| Cities | 0.479 | | 0.276 | 0.5 | 0.674 | **0.74** | 0.722 | 0.651 | 0.62 | 0.556 | 0.508 | 0.489 |
| People | 0.677 | | 0.551 | 0.645 | 0.727 | 0.8 | 0.857 | **0.863** | 0.857 | 0.84 | 0.778 | 0.712 |
| Colors | 0.638 | | 0.706 | 0.811 | **0.952** | 0.889 | 0.857 | 0.83 | 0.8 | 0.786 | 0.733 | 0.677 |
| Misc(other) | 0.589 | | 0.568 | 0.706 | 0.788 | 0.867 | **0.881** | 0.867 | 0.848 | 0.803 | 0.763 | 0.702 |
| Average | 0.5822 | | 0.5732 | 0.7008 | 0.8136 | 0.8402 | 0.8402 | 0.819 | 0.8018 | 0.7658 | 0.7148 | 0.6426 |
| Combined | 0.56 | | 0.541 | 0.674 | 0.785 | 0.824 | **0.828** | 0.793 | 0.768 | 0.719 | 0.669 | 0.617 |

Table 7 expands on the results of the combined group by including precision and recall metrics as well as the counts of correct and incorrect (inaccurate) candidate URIs, i.e., identity links *after* filtering. A noticeable improvement is observed from the original (baseline) F-score of 0.560 with the only drop at the $\delta \geq 0.9$ level (0.541). This drop is caused by high precision (0.898 versus 0.389) but low recall (0.388 versus 1). Nonetheless, this threshold may still be desirable as it resulted in only 7 errors remaining from the initial 251 inaccuracies. Also shown is our apparent optimal threshold of $\delta = 0.5$ preserving 200 sameAs candidates of which 149 were true positive and only 51 false positive classifications. If high recall is desired even a low threshold of 0.0 to 0.2 would result in F-score improvement over the original (non-filtered) resource set.

In Fig. 2 we provide a scatter plot of F-score versus disambiguation ratio for the complete dataset of 411 URIs, i.e., identity link candidates with oracle identified correct/incorrect entries shown in blue (true positive) and red (true negative). The plot displays the desired characteristic that inaccurate links trend towards the lower-scoring disambiguation ratio while correct links gravitate toward higher-scoring values.

Our experimentation reveals that SCID can identify a significant number of erroneous identity links independent of any specific topic (animals, cities, etc.).

A user-specified threshold of 0.5 to 0.6 appears to work well as a general purpose setting for best F-score (Fig. 2 and Tables 6 and 7).

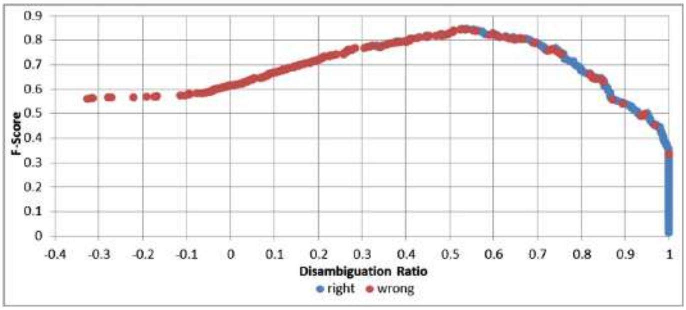


**Fig. 2.** Scatter plot of F-score versus disambiguation ratio for combined dataset with oracle-identified right(blue) and wrong(red) identity links (Color figure online).

**Table 7.** Precision, Recall, F-score statistics for the combined (non-grouped) set with correct/incorrect counts of the candidate URIs (i.e., identity links) after filtering.

| | Precision | Recall | F-Score | Correct | Incorrect |
|---|---|---|---|---|---|
| Original | 0.389 | 1.000 | 0.560 | 160 | 251 |
| δ≥0.9 | 0.899 | 0.388 | 0.541 | 62 | 7 |
| δ≥0.8 | 0.841 | 0.563 | 0.674 | 90 | 17 |
| δ≥0.7 | 0.832 | 0.744 | 0.785 | 119 | 24 |
| δ≥0.6 | 0.789 | 0.863 | 0.824 | 138 | 37 |
| δ≥0.5 | 0.745 | 0.931 | 0.828 | 149 | 51 |
| δ≥0.4 | 0.677 | 0.956 | 0.793 | 153 | 73 |
| δ≥0.3 | 0.626 | 0.994 | 0.768 | 159 | 95 |
| δ≥0.2 | 0.561 | 1.000 | 0.719 | 160 | 125 |
| δ≥0.1 | 0.503 | 1.000 | 0.669 | 160 | 158 |
| δ≥0 | 0.446 | 1.000 | 0.617 | 160 | 199 |

## 7 Conclusion

In this paper, we proposed a technique (SCID) to discover coreference inaccuracies in existing sameAs links. Experimental results indicate that SCID can improve the quality of an identity collection by correctly flagging questionable identity assertions. A distinguishing feature is that unlike most existing approaches (e.g., [2, 7]), SCID considers the semantics of the resources associated through identity links. The semantic relatedness and disambiguation ratio scores could provide a quantitative measure of semantic similarity for those seeking more than a binary correct/incorrect classification. This is an important advantage over the existing work that leads to future experimentation with skos:closeMatch, skos:exactMatch, and owl:equivalentClasses as the identity links. Furthermore, unlike some existing work addressing the same problem, our proposal does not depend on domain-specific rules that have to be defined by human experts (as e.g. in [12]). We position SCID not as a replacement for such rules but as a supplemental verification step.

Development plans for SCID include improvements to the relevant keywords selection method (Algorithm 1) and the exploration of alternative vector similarity methods such as Cosine similarity, Euclidean distance, and SVR/SVM (Algorithm 2).

## References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia-A crystallization point for the Web of Data. Web Semant.: Sci., Serv. Agents World Wide Web **7**(3), 154–165 (2009)
2. de Melo, G.: Not quite the same: Identity constraints for the Web of Linked Data. In: des Jardins, M., Littman, M.L. (eds.) Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Menlo Park, CA, USA. AAAI Press (2013)
3. Gianluca, D., Difallah, D.E., Cudré-Mauroux, P.: ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: Proceedings of the 21st International Conference on World Wide Web, pp. 469–478. ACM (2012)
4. Ding, L., Shinavier, J., Shangguan, Z., McGuinness, D.L.: SameAs networks and beyond: analyzing deployment status and implications of owl:sameAs in Linked Data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 145–160. Springer, Heidelberg (2010)
5. Ferrara, A., Nikolov, A., Scharffe, F.: Data linking for the semantic web. Int. J. Semant. Web Inf. Syst. **7**(3), 46–76 (2011)
6. Ferrara, A., Nikolov, A., Scharffe, F.: Data linking for the semantic web. Semant. Web: Ontology Knowl. Base Enabled Tools, Serv., Appl. **2013**, 169 (2013)
7. Guéret, C., Groth, P., Stadler, C., Lehmann, J.: Assessing linked data mappings using network measures. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 87–102. Springer, Heidelberg (2012)
8. Hogan, A., Polleres, A., Umbrich, J., Zimmermann, A.: Some entities are more equal than others: Statistical methods to consolidate linked data. In: Proceedings of the Workshop on New Forms of Reasoning for the Semantic Web: Scalable & Dynamic (NeFoRS2010) @ ESWC2010 (2010)
9. Hu, W., Chen, J., Qu, Y.: A self-training approach for resolving object coreference on the semantic web. In: Proceedings of the 20th International Conference on World Wide Web (WWW 2011), pp. 87–96. ACM, New York (2011)
10. Maali, F., Cyganiak, R., Peristeras, V.: Re-using cool URIs: entity reconciliation against LOD hubs. In: Proceedings of the Linked Data on the Web (LDOW 2011)
11. Datasets for the Identity Recognition Task. Instance Matching Track of the Ontology Alignment Evaluation Initiative 2014 Campaign. http://islab.di.unimi.it/im_oaei_2014/index.html
12. Papaleo, L., Pernelle, N., SaÏs, F., Dumont, C.: Logical detection of invalid SameAs statements in RDF data. In: Janowicz, K., Schlobach, S., Lambrix, P., Hyvönen, E. (eds.) EKAW 2014. LNCS, vol. 8876, pp. 373–384. Springer, Heidelberg (2014)

13. Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the linked data best practices in different topical domains. In: Mika, P., et al. (eds.) ISWC 2014, Part I. LNCS, vol. 8796, pp. 245–260. Springer, Heidelberg (2014)
14. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Trans. Knowl. Data Eng. **25**(1), 158–176 (2013)
15. Halpin, H., Herman, I., Hayes, P.: When owl:sameAs isn't the same: an analysis of identity links on the semantic web. In: Linked Data on the Web (LDOW 2010)