Detecting errroneous sameAs links using mathematical programming

Sixiao ZHU
sixiao.zhu@telecom-paristech.fr

March 15, 2021

Contents

Preface

I would like to thank my supervisor Nathelie Pernelle for her excellent guidance during this process.
I also wish to thank Mme. Fatihna Sais, Mme. Dominique Quadri, Mme. Yue Ma, and Mr. Joe Raad for their supports.

Chapter 1

Introduction

The web of Linked Data will continue to grow exponentially. More and more independently developed data sets are added to LOD, inevitably different data sets contains entries describing the same thing, only with different label, textual description, and ways to name predicates. we need to identify these "essentially the same" entries by linking them with a remark to rich more information from different data sets. The most common used in linked open data is the "sameAs" predicate. Some sameAs links in existing LOD data sets are erroneous for various reasons, efforts have been put into detecting such erroneous link, in this work we design a new method that has the advantage of being able to aggregate existing methods.

Chapter 2

Context of the work

2.1 Semantic Web

Semantic Web [1] is the extention of data on the web in a way that, in addition to having meaning for humans, also have meaning for computers. This is achieved by adding semantic labels to non-structured data, which makes data able to be reasoned and share by computers. The significance of semantic web is, By W3C, "providing a common framework that allows data to be shared and reused across application, enterprise, and community boundaries"

2.2 Linked Open Data and Lod-a-lot

Linked data is an important concept in semantic web. The aim of Linked Data [2] is to relate existing data described using the RDF model (Resource Description Framework) so that machines can browse the Web. The semantic interlinking of linked data makes it possible for semantic queries. Linked Open Data (LOD) [3] is linked data in the spirite of open accessing, in other words, data in LOD is released under an open licence.
LOD-a-lot [4] is a crawling of existing data sets in LOD, it means to offer lowcost consumption of a large portion of LOD.

2.3 sameAs link

The "sameAs" is a RDF predicate indicating that the two entries concerned by this predicate discribe the same physical world entity. It is a key part of the Semantic Web, as determine whether two entries represents the same thing appears as a common task in applications. The extrinsic Logical sense of "x sameAs links to y" is that x and y coincide on every predicates applied on them.

2.4 Detection of erroneous sameAs link

In the construction of Linked Open Data, an essential task is to match entries representing the same entities in independently developed different data sets by sameAs links. Due to the large scale of LOD, it is unevidable that some of sameAs links are mislabeled, these are some of the major reasons for mislabeling:

- A large amount of sameAs links are generated by automatic tools, which is inherently not perfect in their precision.
- Entries in the LOD are written in different languages, which introduces translation confusion.
- There are entries are philosophically undecidable whether they correspond to the same entity.

2.5 Pevious method

In this section we introduce some of the approaches taken by other reseachers.

2.5.1 Source Trustworthiness

Source Trustworthiness is an early approach for detecting erroneous identity statements in the Web of Data, an representative of such approach is [5], it hypothesizes that links published by trusted sources are more likely to be correct, by exploiting the transitivity nature of sameAs link, the solver detects logic conflicts against information issued from trusted sources.

2.5.2 UNA or Ontology Axioms Violation

We say a data set possesses the Uniaue Name Property (UNP) if each entity admits a unique entry in this data set. It is common (and resonable) that all independently developed knowledge base data set possesses this property. If an entry in data set A with UNP is sameAs-connected to more than one entries of another data set B with UNP, we can assert that only one of these sameAs links are connect, as identity mappings between A and B should be injective, we call this case Unique Name Violation.

2.5.3 Functional property (FP)

Some properties are functional, for example, birth date, if entries a and b are different on birth date property, a and b refer to different persons.

2.5.4 Content-based

This is a straight forwad approach that model our problem as a binary classification problem, it directly compares the contents of each pair of entries and determine with certain criterion whether they corresponds to the same entity, and as such determine whether the sameAs link (if exists) is erroneous. A possible approach [6] is to model each entry as a feature vector in a high dimensional vector space and assign a score to each sameAs links.

2.5.5 Network topology based

This approach explores the statistical correlation between the likeliness of being erroneous and the local network topology, that is, if a cluster of entries are densely interconnected, it is probable that members of this cluster corresponds to the same entity as is been declare by large amounts of links, and in return links between these members are likely to be correct. It is a mechanism of mutual enforcement of confidence. Work of Joe Raad [7] exploits this mechanism.

Chapter 3

Problem formalization

3.1 Definition of terms used

3.1.1 Collection of Entries

Definition 1 (Entity). *An Entity is a thing with distinct and independent existence in the physical world. Denoted $\mathcal{E}$*

Definition 2 (Entry). *An entry is the correspondence of an Entity in a dataset.*

Different data set could each have their own entry corresponding to the same entity. An entry is expressed in the form of an URI (Uniform Resource Identifier).

Definition 3 (Triple). *A triple is the basic element of a knowledge graph, expressed in the form <Subject, Predicate, Object>, the predicate is the semantic link between subject and object (two entries).*

The "label" is an example of a predicate, its associated object is the briefest description of its subject.

Definition 4 (Independent Data Set). *An independent data set is simply a collection of triples.*

Definition 5 (Unique Name Assumption). *Unique Name Assumption (UNA) asserts the fact that each entity will at most has a single appearance as an entry in an independent data set.*

Definition 6 (Linked Data). *An linked data set is a the aggregation of multiple independent data set with extra triples describing semantics relations between entries from different independent data set. A important example of such semantics relation is the "sameAs" link. The data set "lod-a-lot"[4]*

The data set "lod-a-lot"[4] is a example of a linked data set

Definition 7 (Content of an entry). *The content of an entry e is all the triples in the linked data set with subject being e, it could be thought of the knowledges related to the entity that e represents.*

3.1.2 SameAs Graph

Definition 8 (SameAs Graph)**.** *The SameAs Graph denoted* (N, E) *is an undirected graph of entries. We have* N *a subset of entries of the linked data set, and* E *set of undirected edges.* $\forall x, y \in N, \{x, y\} \in E$ *if and only if triple* $<x$*, sameAs,* $y>$ *or* $<y$*, sameAs,* $x>$ *appears in the linked data set.*

Definition 9 (Equivalent Class)**.** *A Equivalent Class is a connected component of the SameAs Graph*

The equivalent class will be the basic object we will operate on. Our goal of this work is for every equivalent class (N, E), select a subset of edges $F \subset E$ as erroneous sameAs links.

3.2 Mathematical programming based approach

In this work we propose a mathematical programming based approach to the goal proposed in previous section. In the process of deciding the validaty of a samsAs link, multiple highly heterogeneous factors would be taken into account, these factors will in a sense "compete", that is, the result given by one decision factor could be reversed by another. For example, a sameAs link between two entities with high textual similarity could be firmly classified erroneous by the UNA decision factor. It is therefore crucial to adopt a framework that is flexible enough to aggregate different decision factors. Mathematical programming would be a fine choice for our need, as content based factor could be modeled as optimization object, while UNA and Source Trustworthiness factor could be naturally modeled as constrains.

3.2.1 Measurement of content similarity

In order to use content of an entry as an decision factor, we need to have a measurement on the similarity of entries. Several methods exists to serve this purpose:

- Model each entry as a vector of large dimension, where each predicate corresponds to one dimension. The problem is this approach is that lots of work remains to be done for effective embedding of ontology entries as high dimensional vectors.
- Model each entry as a big of words. The problem of this approach is that the distribution of information is rather non-balanced among all the words, for example, for two person related entries, a difference on birth date is sufficient to distinguish the two entries, as well the sameness on birth date rend strong confidence that two entries describe the same person, thus words for birth date is much more informative than other descriptive words.
- Embedding the textual description (specifically "resume") of entries into a Euclideans space, and use measurements in Euclidean space, such as cosine similarity or euclidean distance, as measurements of entries. The problem of this approach

is that a large amounts of entries in our data set are short in textual length, the embedding result of these short entries will be unstable.

Our choice of measurement this work is probably the simplest one possible: The edit distance of the entry labels. We adopt this because by our observance of the data set, for the majority of the equivalent classes, the obvious texual disparity of entry label is enough to deliver sufficient information to identify many misconnected entries.

3.2.2 Language issue

In our data set, it is common case that an entity appears as several different entries in different language. Content based similarity measurements can not be directly applied to such heterogeneous environment. A possible work around is:

- when it comes to measuring content similarity, we only take English entries into account.
- Only consider content similarity for entries in the same language, which is a extending of the previous one.

For now we choose the first approach for simplicity reason, this gives us the formal definition of content similarity measure.

Definition 10 (Content Distance between entries)**.** *For any e_1 and e_2 pair of entries in the sameAs graph, we define their content distance as*

$$d(e_1, e_2) = \begin{cases} levenshtein(label(e_1), label(e_2)) & \text{if } e_1 \text{ and } e_2 \text{ are english nodes} \\ 0 & \text{if not} \end{cases}$$

where $levenshtein(label(e_1), label(e_2))$ means denotes the Levenshtein distance [8] between label of e_1 and label of e_2

3.2.3 Transtivity of samesAs relation and the criteria for a cut

The content based decision factor gives us information of the extent of misconnecting of any two entries, it still remains to choose which links should be decide erroneous. For simplicity of describing we define term for this operation:

Definition 11 (Cut)**.** *A cut of a equivalence class (N, E) is the operation of selecting of a subset of edges $F \subset E$, declaring links in F as erroneous, and deleting F from E.*

Definition 12 (Relation of path connectivity)**.** *For any two entries x and y in a equivalent class, we denote $x \sim y$ if there exists at least one path from x to y.*

$\sim$ is naturally reflective and symmetric, what is of specific importance is that it is transitive. A equivalent class is by definition fully connected, however, performing a cut operation could result several separated connected components, each connected

component would be interpreted as representing the same entity, in other words, we assume entries in the same component are essentially the same to each other, thus in the computing of content similarity, entries in the same component after cutting should in a sense "work together". This principle may seem trivial, however it reflects the trade off in the process of mathematical programming, x. Since $x \sim y$ is interpreted as x is essentially same as y, keeping $x \sim y$ and $y \sim z$ will lead to $x \sim z$, thus admitting x is essentially the same as z. We have the following distance between connect components after cutting.

Definition 13 (Distance between connected components)**.** *For two connected components C_1 and C_2, we define the distance between them as*

$$d(C_1, C_2) = \sum_{e_1 \in C_1, e_2 \in C_2} d(e_1, e_2)$$

A intuitive principle concerning the choice of a cut is that it better to make the resulting connected classes "as far as possible" in the sens of content similarity, since each connected component is interpreted as corresponding to an entity. Thus we have the following principle:

Principle (Maximum discrepancy principle)**.** *The preferable cut would leave the maximum average distance between connected components.*

If we cut all the edges in a equivalent class, that is to leave each entry totally separated from each other, we would have a average component distance down to 0, but that is obviously unacceptable as it can not be the case that every sameAs link is false, thus we want to have a limitation of number of cut, and let the mathematical programming solver to find the best cut under this limitation.

3.2.4 MP program formalization

Suppose $G = (N, E)$ an equivalent class, where N set of entries and E set of edges in the form of unordered pairs. Let $M \subset N$ be the English entries amoung N. We denote $\forall i \in N, V_i$ as neighbour of entry i. We have as decision variables:

- r_{ij} where $i, j \in N$ denotes reachability (in the sense of path connectivity) from entry i to entry j, taking value from $\{0, 1\}$. $r_{ij} = 0$ means entry i could reach entry j in a possible cutting setting.
- x_{ij} where $\{i, j\} \in E$ denotes whether we cut edge $\{i, j\}$. $x_{ij} = 1$ means we cut $\{i, j\}$, 0 means not.

We have has precomupted quantities

- d_{ij} denotes the distance (measure of difference) between entries i and j, the lower d_{ij}, the similar are the two entries. Note that if i or j is non-English entry, d_{ij} is set to zero

We have as objective:

$$Minimize \sum_{i,j \in N} d_{i,j} * r_{i,j}$$

We have as contrains:

$\forall i \in N, r_{ii} = 1$	Self Reflectiveness
$\forall i, j \in N, \forall k \in V_j$, $r_{ij} - r_{ik} + 2x_{jk} \geq 0$, $r_{ik} - r_{ij} + 2x_{jk} \leq 0$	Transtivity (*)
$\sum_{\{i,j\} \in E} x_{ij} \leq LIMIT$	Cut number limit
If $\{i, j\} \in E$ but determined erroneous by UNA, $x_{ij} = 1$	UNA decision factor
If $\{i, j\} \in E$ but determined errorneous by FP, $x_{ij} = 1$	FP decision factor

(*) If x_{jk} is set to 1, r_{ij} and r_{ik} can freely change withnot violating the constrain. If x_{jk} is set to 0, means we reserve the edge between k and j, the reachability of i to j has be the same as i to k, in this case, the constrain enforces that $r_{ij} \geq r_{ik}$ and $r_{ik} \geq r_{ij}$, thus $r_{ij} = r_{ik}$.

Chapter 4

Experiments

4.1 Setup

4.1.1 Solver

We have used Gurobi [9] as our mathematical programmig solver. The problem scale is typically less then 10,000 variables and constrains, thus can be effectively solved.

4.1.2 Data storing

Lod-a-lot data set contains 28,362,198,927 Triples, and takes 524GBs to store. It is orginized in the HDT (Header, Dictionary, Triples [10])format.
The work [11] has developped data set consisted only by sameAs links by crawling the Lod-a-lot set, grouped by equivalent classes. It contains 558,943,166 sameAs links, grouped and indexed by equivalent class.
Figure 4.1 shows typical topological structure of equivalent classes. We use a computer with Intel(R) Xeon(R) E5-2630 v4 @ 2.20GHz CPU and 125GB memory as our server, a portable hard drive as storage of Lod-a-lot data set and sameAs equivalence class data set.

4.1.3 Data Washing

We do two rounds of data washing:

- Washing out entries containing unrecognizable caractors.
- Some entries are redirected to others, typically these entries come with sameAs links. We consider by default entries linked with redirection links are essentially the same, thus if we have entry a sameAs connected to entry b, and entry b redirected to entry c, we delete the entry b and fix the sameAs link by adding a new one from a to c. In our test case "Obama" equivalent class, 179 entries out of total 439 entries are redirected to other entries.

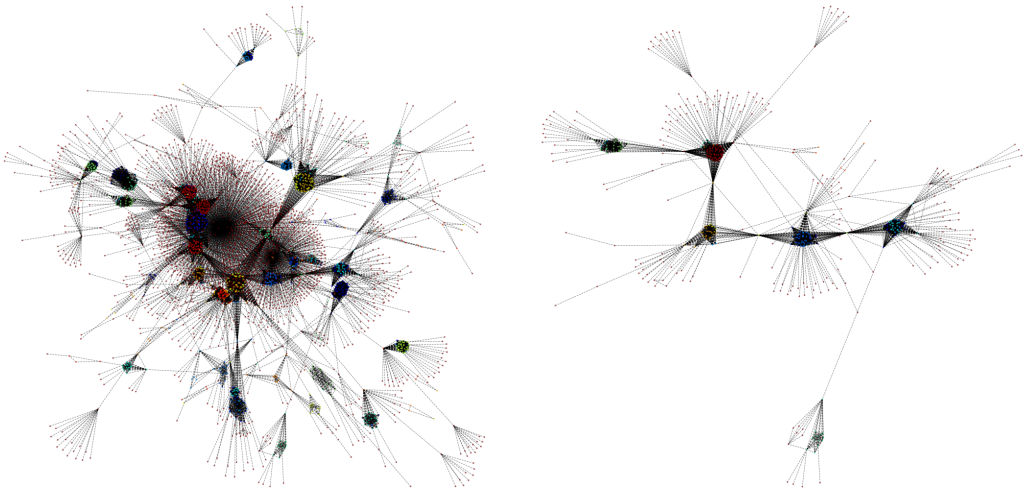

Figure 4.1: Typical topological shape of equivalent classes

4.1.4 Experiment design

For the limitation of timing, we have just used as content similarity in the mathematical programming, UMA and FP factors can be easily added into the framework by adding simple constrains
We have tested our method on two equivalence class (golden standard) constructed by human labeling, we evaluate our method by precision and recall of the result obtained. Golden standard the "Obama Set", contains 439 entries and 7569 edges, its entries corresponds to one of:

- Barack Obama
- Precidency of Obama
- Senator career of Obama
- President transation of Obama
- ...

Golden standard 2 contains contains 236 entries and 4645 edges, its entries corresponds to one of:

- Pripyat City (A Ukraine city)
- Pripyat River

We show in Table 4.1 solver's time consumptions in different data set and different limit.

We show our precision/recall result of Obama set as in Table 4.2, and visualized in Plot 4.2

Data set name / Limit of cut	10	20	30
Obama set	98	309	3561s
Pripyat set	146	1	1

Table 4.1: Time consumption

Data set name / Limit of cut	10	20	30
Precision	4/10	6/20	6/30
Recall	4/17	6/17	6/17

Table 4.2: Performance

For the Pripyat set, the cut found by the solver have no overlap with erroneous links accroding to labeling (in this case we have precision and recall both to zero), several reasons for this rather strange result:

- The golden standard is rather problematic, some erroneous links found by the solver is not declare by the golden standard, as some entries is not labeled by the golden standard.
- In this equivalence class only 10% of the entries are in English, information missed on some crucial points could cause great fluctuation of performance. Figure 4.3 shows the cutting result of Pripyat set, large red nodes reprensent English entries reprsents "Pripyat the city", large yellow nodes represents "Pripyat the river", other small blue nodes are non-English entries, red edges are the sameAs links that have been cut buy the solver. Nodes with black circle is "pivot" point in a sense, its contextual information is of particular importance of correctly classifying nearby nodes, yet in the these two entries are not in English, thus not used in the objective function.

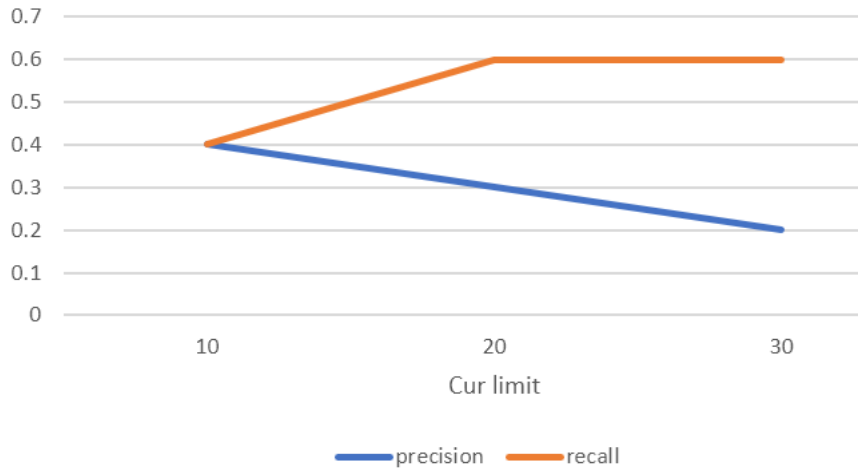

Figure 4.2: Precision / Recall of Obama set

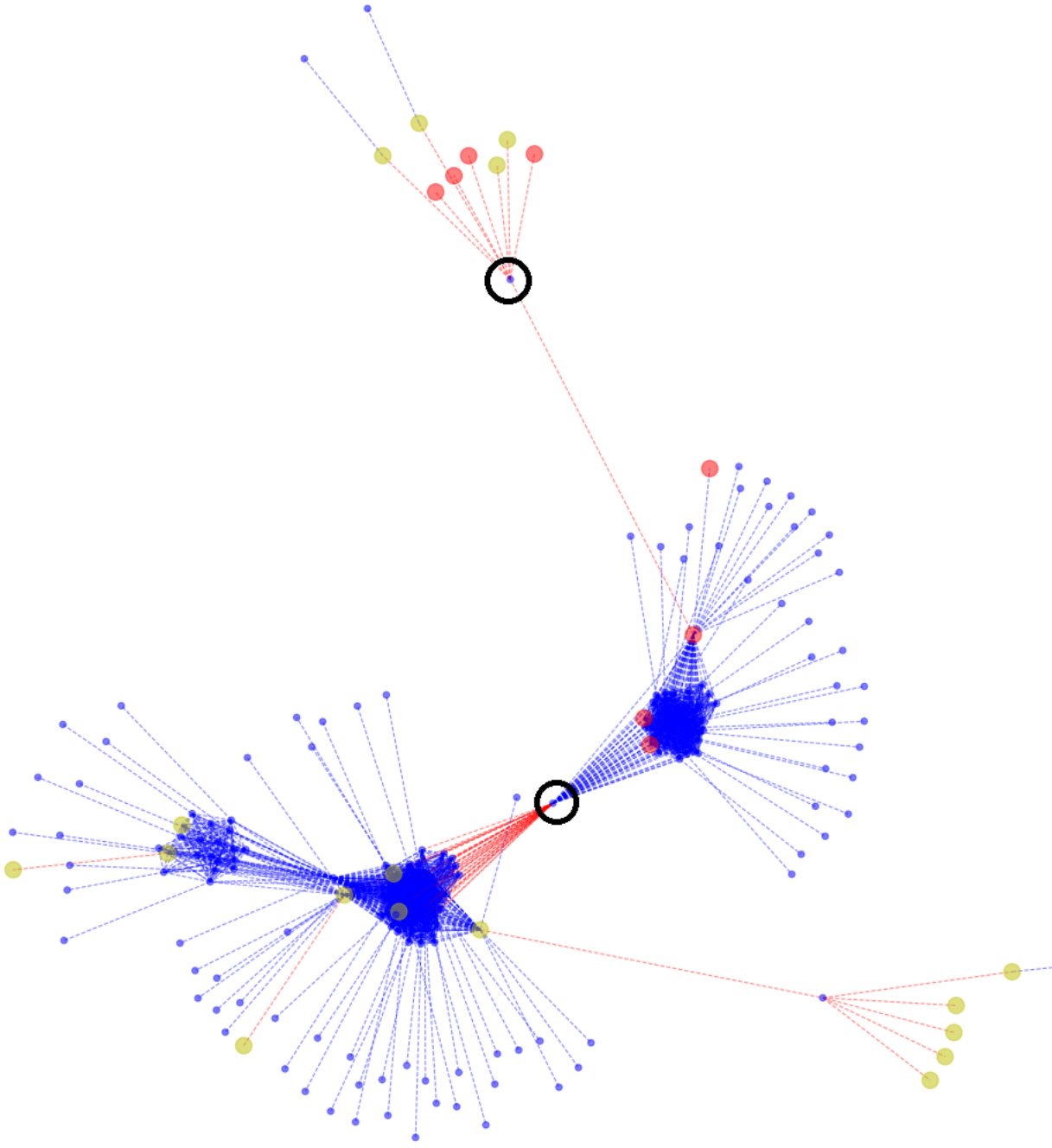

Figure 4.3: Caption

Chapter 5

Conclusion and future improvement

In this work we proposed a mathematical programming based apporach to detect erroneous sameAs links. There are several directions for future improvement:

- We can use more sophisticated distance function.
- Right now we have only taken distances between English entries into objective function, however edit distance makes sense as long as the two entries concerned are in the same language, not necessarily in English.
- In the experiment we have not added in UMA decision factor and functional property decision factor, which are have been widely tested effective in practice.

Bibliography

[1] *Semantic Web*. URL: https://en.wikipedia.org/wiki/Semantic_Web.

[2] *Linked data*. URL: https://fr.wikipedia.org/wiki/Linked_data.

[3] *Linked Open Data*. URL: https://fr.wikipedia.org/wiki/Linked_open_data.

[4] Javier D. Fernández et al. "LOD-a-lot - A Queryable Dump of the LOD Cloud". In: *International Semantic Web Conference*. 2017.

[5] Philippe Cudré-Mauroux et al. "idMesh: graph-based disambiguation of linked data". In: *Proceedings of the 18th international conference on World wide web*. ACM. 2009, pp. 591–600.

[6] Heiko Paulheim. "Identifying Wrong Links between Datasets by Multi-dimensional Outlier Detection". In: *WoDOOM*. 2014.

[7] Joe Raad et al. "Detecting Erroneous Identity Links on the Web Using Network Metrics". In: Sept. 2018, pp. 391–407. ISBN: 978-3-030-00670-9. DOI: 10.1007/978-3-030-00671-6_23.

[8] *Levenshtein Distance*. URL: https://en.wikipedia.org/wiki/Levenshtein_distance.

[9] *Gurobi*. URL: https://www.gurobi.com.

[10] *Header Dictionary Triples*. URL: http://www.rdfhdt.org/what-is-hdt/.

[11] Wouter Beek et al. "sameAs.cc: The Closure of 500M owl:sameAs Statements". In: June 2018, pp. 65–80. DOI: 10.1007/978-3-319-93417-4_5.