Pontifícia Universidade Católica
do Rio de Janeiro

Alexander Arturo Mera Caraballo

Clustering and Dataset Interlinking Recommendation in the Linked Open Data Cloud

Tese de Doutorado

Thesis presented to the Programa de Pós-graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Ciências - Informática.

Advisor: Prof. Marco Antonio Casanova

Rio de Janeiro
March 2017

Pontifícia Universidade Católica
do Rio de Janeiro

Alexander Arturo Mera Caraballo

Clustering and Dataset Interlinking Recommendation in the Linked Open Data Cloud

Thesis presented to the Programa de Pós-graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Ciências - Informática. Approved by the undersigned Examination Committee.

Prof. Marco Antonio Casanova
Advisor
Departamento de Informática – PUC-Rio

Prof. Antonio Luz Furtado
Departamento de Informática – PUC-Rio

Prof. Bernardo Pereira Nunes
Departamento de Informática – PUC-Rio

Prof. Giseli Rabello Lopes
UFRJ

Prof. Luiz André Portes Paes Leme
UFF

Prof. Marcio da Silveira Carvalho
Vice Dean of Graduate Studies
Centro Técnico Científico – PUC-Rio

Rio de Janeiro, March 17th, 2017

Alexander Arturo Mera Caraballo

Alexander Arturo Mera Caraballo holds a master in computer science degree from Pontifical Catholic University of Rio de Janeiro (PUC-Rio), also a system engineering degree from University of Nariño (UDENAR). Alexander worked as a system analyst at the Central Coordination for Distance Learning (CCEAD) of PUC-Rio. In addition, he also worked as a system analyst at the Central Coordination for Planning and Evaluation (CCPA) of PUC-Rio. His main research topics areas include Semantic Web, Information Retrieval, Multimedia Content, Information Extraction and Natural Language Processing.

Bibliographic data

Mera Caraballo, Alexander Arturo

Clustering and Dataset Interlinking Recommendation in the Linked Open Data Cloud / Alexander Arturo Mera Caraballo; advisor: Marco Antonio Casanova. – 2017.

89 f.: il. ; 29,7 cm

1. Tese (Doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui bibliografia.

1. Informática – Teses. 2. Recomendação de conjuntos de dados para interligação. 3. Dados Interligados. 4. Web Semântica. I. Casanova, Marco Antonio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CCD: 004

I dedicate this thesis to people whom I love most in the world:

My grandparents, my parents, my brother and Dana.

Acknowledgments

First of all, I would like to express my deep gratitude to my advisor, Prof. Marco Antonio Casanova, who before being my advisor was my teacher in a couple of disciplines. These fantastic classes show me the way that I have to follow. During all this time, he supported me and guided me through this new world of research. Casanova characterized for being an advisor that cares for his students, for this reason, he always have time, always with an excellent humor, great attitude, and brilliant ideas. For all these reasons, I am really happy in closing this chapter of my academic career under his mentoring.

I also would like to thank to Prof. Bernardo Pereira Nunes, who is a great colleague and friend. I remember that, we started to write our first paper in the library of our university. Since then, we have written and published many more. His friendship, knowledge and mentoring helped me to improve and engage with my research.

I am also thankful with Prof. Giseli Rabello Lopes and Prof. Luiz André Portes Paes Leme, who are always available to help and collaborate. Thanks for always share your great ideas and taking your time of your busy schedule to spend long hours in writing papers.

Thanks to all member of CCEAD PUC-Rio, for being so supportive. They provide me an excellent environment that helps me to achieve personal as well as

professional goals. Thanks for all opportunities, advice and to make this enjoyable experience.

I also would like to thank to all member of CCPA PUC-Rio, for being great colleagues. They provide me an excellent environment that helps me to achieve personal as well as professional goals. Thanks for all opportunities, advice and to make this enjoyable experience.

Thanks for my colleague Jose Eduardo Talavera, for sharing his ideas, and drinking a coffee always that we have a break.

Thanks to CAPES, FAPERJ and PUC-Rio, for the grants that made this research possible.

Last but not the least important, I owe more than thanks to my family. This work would not been possible without their support, comprehension, motivation, trustfulness and love. Thanks for always being in my side and for being part of this dream that I am making real.

Abstract

Mera Caraballo, Alexander Arturo; Casanova, Marco Antonio (Advisor). **Clustering and Dataset Interlinking Recommendation in the Linked Open Data Cloud**. Rio de Janeiro, 2017. 89p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The volume of RDF data published on the Web increased considerably, which stressed the importance of following the Linked Data principles to foster interoperability. One of the principles requires that a new dataset should be interlinked with other datasets published on the Web. This thesis contributes to addressing this principle in two ways. First, it uses community detection algorithms and profiling techniques for the automatic creation and analysis of a Linked Open Data (LOD) diagram, which facilitates locating datasets in the LOD cloud. Second, it describes three approaches, backed up by fully implemented tools, to recommend datasets to be interlinked with a new dataset, a problem known as the dataset interlinking recommendation problem. The first approach uses link prediction measures to provide a list of datasets recommendations for interlinking. The second approach employs supervised learning algorithms, jointly with link prediction measures. The third approach uses clustering algorithms and profiling techniques to produce dataset interlinking recommendations. These approaches are backed up, respectively, by the TRT, TRTML and DRX tools. Finally, the thesis extensively evaluates these tools, using real-world datasets, reporting results that show that they facilitate the process of creating links between disparate datasets.

Keywords

Dataset Interlinking Recommendation; Linked Data; Semantic Web.

Resumo

Mera Caraballo, Alexander Arturo; Casanova, Marco Antonio. **Clusterização e Recomendação de Interligação de Conjunto de Dados na Nuvem de Dados Abertos Conectados.** Rio de Janeiro, 2017. 89p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

O volume de dados RDF publicados na Web aumentou consideravelmente, o que ressaltou a importância de seguir os princípios de dados interligados para promover a interoperabilidade. Um dos princípios afirma que todo novo conjunto de dados deve ser interligado com outros conjuntos de dados publicados na Web. Esta tese contribui para abordar este princípio de duas maneiras. Em primeiro lugar, utiliza algoritmos de detecção de comunidades e técnicas de criação de perfis para a criação e análise automática de um diagrama da nuvem da LOD (Linked Open Data), o qual facilita a localização de conjuntos de dados na nuvem da LOD. Em segundo lugar, descreve três abordagens, apoiadas por ferramentas totalmente implementadas, para recomendar conjuntos de dados a serem interligados com um novo conjunto de dados, um problema conhecido como problema de recomendação de interligação de conjunto de dados. A primeira abordagem utiliza medidas de previsão de links para produzir recomendações de interconexão. A segunda abordagem emprega algoritmos de aprendizagem supervisionado, juntamente com medidas de previsão de links. A terceira abordagem usa algoritmos de agrupamento e técnicas de criação de perfil para produzir recomendações de interconexão. Essas abordagens são implementadas, respectivamente, pelas ferramentas TRT, TRTML e DRX. Por fim, a tese avalia extensivamente essas ferramentas, usando conjuntos de dados do mundo real. Os resultados mostram que estas ferramentas facilitam o processo de criação de links entre diferentes conjuntos de dados.

Palavras-chave

Recomendação de conjuntos de dados para interligação; Dados Interligados; Web Semântica.

Table of Contents

1. Introduction 16

1.1. Motivation and Challenges 16

1.2. Contributions 17

1.3. Organization 18

2. Related Work 19

2.1. Dataset Interlinking Recommendation 19

2.2. Community Detection 21

2.3. Dataset Profiling 22

2.4. Dataset Catalogs 23

3. Automatic Creation and Analysis of a Linked Data Cloud Diagram 25

3.1. Introduction 25

3.2. Background 27

3.2.1. LOD Concepts 27

3.2.2. Communities and Community Detection Algorithms 29

3.2.3. Clustering Validation Measures 30

3.2.4. Dataset Profiling Techniques 31

3.3. An Approach to Automatic Creation and Analysis of a Linked Data Cloud Diagram 32

3.4. Evaluation Setup 33

3.4.1. Construction of the LOD graph and Description of the Ground Truth 34

3.4.2. Setup of the Dataset Clusterization Step 34

3.4.3. Setup of the Dataset Community Description Step 36
3.5. Results 36
3.5.1. Performance of the Dataset Clusterization Step 37
3.5.2. Performance of the Dataset Community Description Step 38
3.6. Discussion and Analysis 40
3.6.1. An Analysis of the Dataset Clusterization Results 40
3.6.2. An Analysis of the Dataset Community Description Results 41
3.7. Conclusion 43

4. Dataset interlinking recommendation 45
4.1. Introduction 45
4.2. TRT- The Dataset Recommendation Tool 46
4.2.1. An Approach to Dataset Interlinking Recommendation 46
4.2.2. TRT Architecture 48
4.2.3. TRT GUI 49
4.2.4. Evaluation Setup 50
4.2.5. Results 52
4.2.6. Conclusions 52
4.3. TRTML - A Dataset Recommendation Tool based on Supervised Learning Algorithms 53
4.3.1. An Approach to Dataset Interlinking Recommendation 53
4.3.2. TRTML Architecture 54
4.3.3. TRTML GUI 55
4.3.4. Evaluation Setup 56
4.3.5. Conclusions 60
4.4. DRX - A LOD Dataset Interlinking Recommendation Tool 61
4.4.1. An Approach to Dataset Interlinking Recommendation 61

4.4.2. DRX Architecture 64

4.4.3. DRX GUI and Case Study 65

4.4.4. Evaluation Setup 71

4.4.5. Discussion 74

4.4.6. Conclusions 78

4.5. Tools comparison 78

4.5.1. Experiment Setup 78

4.5.2. Results 79

4.5.3. Analysis of the Features 80

4.6. Conclusions 81

5. Conclusions and Future Work 83

List of Figures

Figure 1. Community analysis process of the LOD. 33

Figure 2. Local and quasi-local indices. 47

Figure 3. Architecture TRT tool. 49

Figure 4. TRT tool interface. 50

Figure 5. MAP and Recall of the local and quasi-local indices. 52

Figure 6. Link prediction measures. 53

Figure 7. Architecture TRTML tool. 55

Figure 8. Precision of the supervised classifiers by the percentage of (ground truth) unlinked dataset pairs considered (100%, 75%, 50%, 25% and 1%). 58

Figure 9. Recall of the supervised classifiers by the percentage of (ground truth) unlinked dataset pairs considered (100%, 75%, 50%, 25% and 1%). 59

Figure 10. F-measure of the supervised classifiers by the percentage of (ground truth) unlinked dataset pairs considered (100%, 75%, 50%, 25% and 1%). 60

Figure 11. Dataset interlinking recommendation approach. 62

Figure 12. Architecture DRX tool. 64

Figure 13. DRX configuration Web Form. 67

Figure 14. Result of the clustering depicted in a dendrogram. 68

Figure 15. Detailed information of the members of a cluster. 69

Figure 16. List of dataset interlinking recommendations. 70

Figure 17. List of Wikipedia categories from rkb-explorer-newcastle dataset. 70

Figure 18. List of Wikipedia entities from rkb-explorer-newcastle dataset. 71

Figure 19. Strategy 1: Overall Mean Average Precision vs. number 73

Figure 20. Strategy 2: Percentage of datasets vs. Overall Mean Average Precision. 74

List of Tables

Table 1. Number of datasets and linksets per topical domain. 28

Table 2. Top 10 best configurations for EBM by decreasing order of NMI. 35

Table 3. Top 10 best configurations for GCE by decreasing order of NMI. 35

Table 4. Best quality results for the community detection/clustering algorithms. 36

Table 5. Co-occurrence matrix of the GCE result. 37

Table 6. EMI matrix of the GCE result. 38

Table 7. Histograms of top-level categories for each community structure. 39

Table 8. Generated fingerprint for the rkb-explorer-newcastle dataset. 66

Table 9. Performance of the tools. 79

Table 10. Features of the tools. 80

List of Abreviations

LOD	Linked Open Data
RDF	Resource Description Framework
VoID	Vocabulary of Interlinked Datasets
LIMES	Link Discovery Framework for metric spaces
URI	Uniform Resource Identifier
EBM	Edge Betweenness Method
GCE	Greedy Clique Expansion algorithm
COPRA	Community Overlap Propagation Algorithm
NMI	Normalize Mutual Information
EMI	Estimated Mutual Information
MAP	Mean Average Precision
WM	Wikipedia Miner
LDSO	LODStats DataSet vOcabulary
TRT	Tripleset Recommendation Tool
TRTML	A Tripleset Recommendation Tool based on Supervised Learning Algorithms
DRX	A LOD dataset interlinking recommendation tool

1
Introduction

1.1.
Motivation and Challenges

The *Web of Data* emerged as a solution to share and reuse structured data on the Web. Tim Berners-Lee coined this term in 2006 and years later he also defined a set of best practices that would be the basis of the success of the Web of Data. Basically, datasets have to follow a set of four rules to become five stars: use URIs as names for things, use HTTP URIs in order to be accessible, provide useful information for things using RDF, and finally include links to other URIs. The collection of the datasets that follow these rules and that are openly available came to be known as the *Linked Open Data* (LOD) cloud.

Despite that almost a decade has passed, there are still some limitations in order to meet these rules. This is visible mainly in the lack of links among datasets that share data. Here, it is important to highlight that the declaration of these links helps to discover more resources, resulting in more contextual information.

Creating links between resources is not an easy task, since the LOD cloud is composed of a large number of datasets that provide information about a huge variety of domains. For this reason, the research community, as a first attempt to visualize the LOD cloud, built the LOD diagram. However, this diagram is not frequently updated and its design involves a manual process to classify datasets. Therefore, a first challenge is to create an approach to group the datasets in the LOD cloud, together with a description of the groups, in an automatic way.

The understanding of the topology of the LOD cloud is not enough to create links; methods and tools to select datasets to be linked are also required. Indeed, the research community proposed tools to discover links between two datasets, which require some expertise to be configured. A fundamental parameter to set up these tools is the pair of datasets (source and target) to be inspected. Therefore, an additional challenge is to provide tools to obtain recommendation of datasets to be

interlinked. The recommendation of datasets for interlinking facilitates the work of data publishers that need to enrich new datasets by interlinking their underlying new data with resources already published in the LOD cloud.

Most dataset interlinking recommendation techniques typically use metadata that are not related to the content itself, which can cause a low precision in the recommendations. For this reason, it would be helpful to develop methods and tools that consider the content itself to provide dataset interlinking recommendations.

1.2. Contributions

This thesis reports contributions to the LOD dataset clustering and dataset interlinking problems, focusing on providing mechanisms to facilitate meeting the Linked Open Data principles.

Our first contribution is the proposal of a novel approach for the automatic creation and analysis of a Linked Open Data (LOD) diagram. This approach includes an automatic clustering of the LOD datasets into dataset communities that is consistent with the traditional LOD diagram, and an automatic process that generates descriptions of the dataset communities.

This contribution is validated through an experimental evaluation using real-world data. The results show the ability of the proposed process to replicate the LOD diagram and to identify new LOD dataset clusters. Finally, experiments conducted by LOD experts indicate that the clustering process generates dataset clusters that tend to be more descriptive than those manually defined in the LOD diagram.

The second, third and fourth contributions address the dataset interlinking problem. These contributions were implemented in three tools that are described in what follows.

Briefly, the TRT tool analyses the Linked Data network in much the same way as a Social Network. The inputs of TRT are: (i) a Linked Data network $G = (S,C)$; (ii) a target dataset t not in S (intuitively the user wishes to define links

from *t* to the datasets in *S*); and (iii) a target context C_t for *t* consisting of one or more datasets *u* in *S* (intuitively the user knows that *t* can be interlinked with *u*). The output is a ranked list *L* of datasets in *S*. The datasets are ranked using (social network) link prediction measures (such as Common Neighbors, Jaccard coefficient, Preferential Attachment and Resource Allocation).

The TRTML tool also addresses the dataset interlinking problem. Basically, it relies on supervised algorithms (such as Multilayer Perceptron, Decision Trees - J48 and Support Vector Machines) and on link prediction measures that explore a set of features (e.g. vocabularies, classes and properties) available for the datasets found in metadata catalogs. In particular, the supervised learning algorithms are responsible for determining the best set of features for the recommendation task.

The last tool, DRX, has five modules responsible for: (i) collecting data from datasets on the LOD cloud; (ii) processing the data collected to create dataset profiles; (iii) grouping datasets using clustering algorithms; (iv) providing dataset recommendations; and (v) supporting browsing the LOD cloud. We validate our approach through an in-depth evaluation using real-world datasets.

The results reported in this thesis were published in conferences in the areas of Semantic Web and Web Science. Our research in social network analysis and dataset profiling as a way to facilitate the automatic creation and analysis of a linked data diagram was publishes in (CARABALLO *et al.*, 2016). The investigation on the problem of dataset interlinking recommendation resulted in three approaches and implemented three tools, two of them published in (CARABALLO *et al.*, 2013, 2014), and the third one is under revision at present.

1.3. Organization

The remainder of this thesis is structure as follows. Chapter 2 discusses related work. Chapter 3 presents an approach for the automatic creation and analysis of a Linked Open Data cloud diagram. Chapter 4 introduces three approaches implemented in Web applications for dataset interlinking recommendation. Finally, Chapter 5 concludes with a summary of the contributions of the thesis, and directions for future work.

2
Related Work

To facilitate the reading and understanding, this chapter groups related work into four topics: dataset interlinking recommendation, community analysis, dataset profiling and dataset catalogs.

The dataset interlinking recommendation section discusses the different methodologies and data used to generate recommendations of dataset for interlinking. The community analysis section presents studies that resort to community analysis concepts and algorithms to uncover the structure of the LOD. The dataset profiling section describes techniques used to generate high-level representations of datasets. Finally, the dataset catalogs section reviews characteristics of the different repositories of metadata available in the Web.

2.1.
Dataset Interlinking Recommendation

Relatively few studies have been published on this topic, despite been a key factor for improving the quality of the LOD. Most of the existing studies explore the metadata available in dataset catalogs.

For instance, (LEME *et al.*, 2013) created a method based on the naïve Bayes classifier to generate a ranked list of related datasets. The relatedness between datasets was measured using linksets, a set of existing links between datasets, retrieved from the Datahub[1] catalog. Similarly, (LOPES; LEME; *et al.*, 2013) took advantage of linksets to provide dataset interlinking recommendations. They used link prediction measures, from the social network analysis field, to estimate the probability of datasets being interconnected.

[1] https://datahub.io/

(NIKOLOV; D'AQUIN, 2011) investigated the use of a Semantic Web index (Sig.ma) (TUMMARELLO *et al.*, 2010) to identify candidate datasets for interlinking. Sig.ma is queried with text literals extracted from *rdfs:label, foaf:name* and *dc:title* properties from a given dataset to find the most overlapping datasets w.r.t to instances. Instead of using instances, (EMALDI; CORCHO; LÓPEZ-DE-IPINA, 2015) relied on the structural characteristics of datasets using a frequent subgraph mining (FSM) technique to identify and possibly establish links between disparate datasets. FSM is an interesting alternative to provide a more efficient approach as it only uses the most frequent subgraphs from a dataset to perform the analysis.

(ELLEFI *et al.*, 2016) characterized datasets using profiling techniques. They represent each dataset as a text document, and extract a set of schema concept for each dataset. Then, they find schema overlapping by calculating the similarity of the concepts between datasets. Finally, dataset interlinking recommendations are given by calculating a ranking score over the text documents. The cosine measure is applied between document vectors over datasets with respect to a given dataset. Instead of using a set of schema concept labels, in this thesis, we characterize the datasets using more general concepts since datasets may have hundreds of concepts, which makes it difficult to identify the most relevant concepts and therefore the search of the related datasets.

(LIU *et al.*, 2016) introduced a framework that harnesses various similarity measures to produce dataset-interlinking recommendations. In order to compute the probability of two dataset being connected, their framework combines ranking measures that consider the content, links and popularity of the datasets. Additionally, they proposed a collaborative similarity measure. They employed all these metrics to create a model by mean of a learning rank algorithm. Despite implementing several measures, their work depends on the input of the data publisher, that is, the technique is not completely automatic. In this thesis, we propose an automatic technique that do not require any input of the data publisher.

2.2. Community Detection

In some cases, data can be modeled as social networks, which can be analyzed to identify "communities", namely, a subset of nodes that share strong relations. This approach has been adopted to model and analyze the LOD.

For instance, (RODRIGUEZ, 2009a) considered community detection algorithms to identify groups of datasets. Additionally, he manually labeled the uncovered groups with the following categories: Biology, Business, Computer Science, General, Government, Images, Library, Location, Media, Medicine, Movie, Music, Reference and Social.

Note that the main purpose of this thesis is not only to assign labels to clusters of LOD cloud but to automatically identify and generate a more up to date version of the LOD diagram, alleviating the arduous task of data publishers to interlink their datasets, and finding popular vocabularies and others relevant statistics of the actual state of the LOD cloud.

A more up-to-date study was conducted by (SCHMACHTENBERG *et al.*, 2014) in late 2014 showing the increasing adoption of the LOD principles, the most used vocabularies by data publishers, the degree distribution of the datasets, an interesting manual classification of datasets by topical domain (media, government, publications, geographic, life sciences, cross-domain, user generated content and social networking), among others. Note that the labels or topical domains are manually assigned, that is, a domain expert inspects the dataset content in order to be labeled.

Community detection algorithms are crucial to create an automatic method to generate LOD diagrams. Several techniques for identifying communities in graph structures were studied by (FORTUNATO, 2010). Basically, a community is represented by a set of nodes that are highly linked within the community and that have a few or no links to other communities. (FORTUNATO, 2010) also presented techniques to validate the clusters found, which we also adopted in this thesis. (XIE; *et al.*, 2013) also explored community detection algorithms. Unlike Fortunato's work, they also considered in their analysis the overlapping structure

of communities, i.e., when a community (of datasets) belongs to more than one category. From the 14 algorithms examined by (FORTUNATO, 2010), we used the top two best performing overlapping algorithms, GCE and COPRA, in our experiments, as well as a non-overlapping algorithm, which we called the Edge Betweenness Method (GIRVAN; NEWMAN, 2002).

2.3. Dataset Profiling

As community detection algorithms essentially analyze graph structures to find communities, profiling techniques also play an important role in the identification, at a content level, of the relatedness between datasets. Since, these techniques aim at elaborating a concise but comprehensive version of datasets.

For instance, (EMALDI; *et al.*, 2015), based on a frequent subgraph mining (FSM) technique, extracted structural characteristics of datasets to find similarities among them. (LALITHSENA *et al.*, 2013) relied on a sample of extracted instances from datasets to identify the datasets topical domains. Topics were extracted from reference datasets (such as Freebase) and then ranked and assigned to each dataset profile.

Analogously, (FETAHU *et al.*, 2014) proposed an automated technique to create structured topic profiles for arbitrary datasets through a combination of sampling, named entity recognition, topic extraction and ranking techniques. A more generic approach to create profiles on the Web was presented by (KAWASE *et al.*, 2014). Kawase's approach generates a histogram (called *fingerprints*) for any text-based resource on the Web based on the 23 top-level categories of the Wikipedia ontology.

In this thesis, we evaluated and adopted Kawase's technique, which demonstrated to be suitable to determine the topical domain of dataset communities. The drawback of Fetahu's approach in our scenario is the large number of categories assigned to a given dataset, which hinders the identification and selection of the most representative topics of a dataset and, consequently, of a community.

2.4. Dataset Catalogs

A catalog provides metadata of the datasets published in the Web. Generally, metadata is created from data publishers, which register manually information about datasets. However, there is other type of catalogs where metadata is calculated automatically from the data itself.

Datahub[2] catalog provides free access to metadata of hundreds of datasets. This catalog run over the CKAN tool that allows, search for data, register published datasets, create and manage groups of datasets, and get notification of updates from datasets. Data publishers can define dataset properties such as: author, name, descriptions, relationships, license, maintainer, and tags. Additionally, Data publishers can retrieve metadata for further analysis through the CKAN API.

Another catalog that also uses the CKAN tool is Mannheim[3]. This catalog was created after the analysis of the adoption of the best practices of datasets in different topical domains (SCHMACHTENBERG et al., 2014). This catalog includes datasets registered in the datahub catalog and datasets that were crawled. The crawl was perform from URIs contained in the Billion Triple Challenge 2012 dataset and URIs from datasets advertised on the public-lod@w3.org mailing list since 2011. Finally, datasets are built from resources that share the same pay-level domain.

LODStats[4] catalog provides comprehensive statistics from datasets available in data.gov, publicdata.eu and datahub.io data catalogs. This catalog presents a descriptive view of the internal structure (e.g. vocabulary/class/property usage, number of triples, linksets) of the datasets. Statistics are published using the LODStats DataSet vOcabulary (LDSO)[5] vocabulary.

2 http://datahub.io/

3 http://linkeddatacatalog.dws.informatik.uni-mannheim.de/

4 http://stats.lod2.eu/

5 LDSO is published at http://lodstats.aksw.org/ontology/ldso.owl

Considering that the majority of the LOD does not comply publishing guidelines. LOD Laundromat (BEEK *et al.*, 2014) provides a way of remove stains from data without the human intervention. Thus, LOD Laundromat became an entry to a collection of cleaned siblings of existing datasets.

3
Automatic Creation and Analysis of a Linked Data Cloud Diagram

This chapter introduces an approach for the automatic creation and analysis of a Linked Data Cloud diagram.

3.1.
Introduction

The Linked Data principles established a strong basis for creating a rich space of structured data on the Web. The potentiality of such principles encouraged the government, scientific and industrial communities to convert their data to the Linked Data format, creating the so-called Linked Open Data (LOD) cloud.

An essential step of the publishing process is to interlink new datasets with those in the LOD cloud to facilitate the exploration and consumption of existing data. These links are modeled as *linksets*, defined as collections of RDF triples whose subjects and objects are described in different datasets. A linkset is represented by the *void:Linkset* class of the VoID[6] vocabulary. Briefly, some of the properties provided for this class are:

- The *void:target* property is used to name the two datasets. Thus, a linkset has to describe different *void:target's*.
- The *void:subset* property states that a linkset is a part of a larger dataset.
- The *void:linkPredicate* property specifies the type of links that interlink two datasets.

Although frameworks to help create links are available, such as LIMES (NGOMO; AUER, 2011) and SILK (VOLZ *et al.*, 2009), the selection of potential

[6] https://www.w3.org/TR/void/

datasets to interlink with a new dataset is still a manual and non-trivial task. One possible direction to facilitate the selection of datasets to interlink with would be to classify the datasets in the LOD cloud by domain similarity and to create expressive descriptions of each class. Thus, the publisher of a new dataset would select the class closest to his dataset and try to interlink his dataset with those in the selected class.

The *LOD* diagram (JENTZSCH et al., 2011; SCHMACHTENBERG et al., 2014), perhaps the best-known classification of the datasets in the LOD cloud, adopted the following categories: "Media", "Government", "Publications", "Life Sciences", "Geographic", "Cross-domain", "User-generated Content" and "Social Networking". However, the fast growth of the LOD cloud makes it difficult to manually maintain the LOD diagram.

To address this problem, we propose a community analysis of the LOD cloud that leads to an automatic clustering of the datasets into communities and to a meaningful description of the communities. The process has three steps. The first step creates a graph to describe the LOD cloud, using metadata extracted from dataset catalogs. The second step uses community detection algorithms to partition the LOD graph into *communities* (also called *clusters*) of related datasets. Here, it is important to note that the LOD graph is created using only linksets, that is, no dataset content is considered and, hence, datasets are included into communities based on their structural similarity. The last step generates descriptions for the dataset communities by applying dataset profiling techniques. As some of the datasets may contain a large number of resources, only a random sample of each dataset is considered. For each dataset community, this step generates a *profile*, expressed as a vector, whose dimensions correspond to relevance scores for the 23 top-level categories of Wikipedia.

The resulting partition of the LOD graph into communities, with the descriptions obtained, may help data publishers search for datasets to interlink their data as follows. Consider a new dataset *d* to be published as Linked Data; the same profiling technique used in the process we propose may be used to generate a profile for *d*, expressed as a vector. Then, a similarity measure (e.g., cosine-based) may be used to compute the similarity between the profile of *d* and the profile of each dataset community. Finally, the data publisher may receive

recommendations for the community with the highest similarity value. This suggested recommendation process is not the focus of this chapter, but it is one of the major motivations of the approach and tool presented in chapter 4.

The remainder of this chapter is structure as follows. Section 3.2 provide background concepts. Section 3.3 describes our approach. Section 3.4 and Section 3.5 present the evaluation setup and the results of our approach, respectively. Section 3.6 discusses and analyses the results. Finally, Section 3.7 presents the conclusions.

3.2. Background

3.2.1. LOD Concepts

A *dataset* is simply a set *t* of RDF triples. A resource, identified by an RDF URI reference *s*, *is defined in t* iff *s* occurs as the subject of a triple in *t*. Given two datasets *t* and *u*, a *link* from *t* to *u* is a triple of the form *(s,p,o)*, where *s* is an RDF URI reference identifying a resource defined in *t* and *o* is an RDF URI reference identifying a resource defined in *u*. A *linkset* from *t* to *u* is a set of links from *t* to *u*.

The set of RDF datasets publicly available is usually referred to as the LOD cloud.

The *LOD graph (or the LOD network*) is an undirected graph *G=(S,E)*, where *S* denotes a set of datasets in the LOD cloud and *E* contains an edge *(t,u)* iff there is at least one linkset from *t* to *u*, or from *u* to *t*.

A *LOD catalog* describes the datasets available in the LOD cloud. Datahub[7] and the Mannheim Catalog[8] are two popular catalogs. LODStats[9] collects statistics about datasets to describe their internal structure (e.g.

[7] http://datahub.io/

[8] http://linkeddatacatalog.dws.informatik.uni-mannheim.de/

[9] http://stats.lod2.eu/

vocabulary/class/property usage, number of triples, linksets). The LOD Laundromat[10] generates a clean version of the LOD cloud along with a metadata graph with structural data.

A *LOD diagram* is a visual representation of the structure of the LOD cloud. At least three versions of the structure of the LOD cloud are currently available (JENTZSCH et al., 2011; SCHMACHTENBERG et al., 2014) provides the most comprehensive statistics about the structure and content of the LOD cloud (as of April 2014). This version of the LOD cloud comprises 1,014 datasets, of which only 570 have linksets. In total, 2,755 linksets (both in- and out links) express a relationship between the datasets contained in this version of the LOD cloud. The datasets are divided into eight topical domains, namely, "Media", "Government", "Publications", "Life Sciences", "Geographic", "Cross-domain", "User-generated Content" and "Social Networking". The datasets are not uniformly distributed per topical domain: "Government" and "Publication" are the largest domains, with 23.85% and 23.33% of all datasets, respectively; "Media" is the smallest domain, containing only 3.68% of all datasets. Table 1 presents the number of datasets in each topical domain, for which linksets are defined. We highlight that the wide variation of the size among the domains represents an additional challenge to community detection/clustering algorithms (ERTÖZ et al., 2003).

Table 1: Number of datasets and linksets per topical domain.

Topical domain	#Datasets	#Inlinks	#Outlinks
Media	21	55	39
Government	136	271	330
Publications	133	772	862
Geographic	24	171	56
Cross-Domain	40	345	180
Life Sciences	63	144	161
Social Networking	90	912	986
User-generated content	42	85	141

[10] http://lodlaundromat.org

3.2.2.
Communities and Community Detection Algorithms

Let G=(S,E) be an undirected graph and G_C=(S_C,E_C) be a subgraph of G (that is, $S_C \subseteq S$ and $E_C \subseteq E$). Let |s| denote the cardinality of a set s.

The intra-cluster density of G_C, denoted $\delta_{int}(G_C)$, is the ratio between the number of edges of G_C and the number of all possible edges of G_C and is defined as follows:

$$\delta_{int}(G_C) = \frac{|E_C|}{|S_C|.(|S_C|-1)/2}$$

Let $\gamma(G_C)$ denote the set of all edges of G that have exactly one node is in S_C. The inter-cluster density of G_C, denoted $\delta_{ext}(G_C)$, measures the ratio between the cardinality of $\gamma(G_C)$ and the number of all possible edges of G that have exactly one node is in S_C and is defined as follows:

$$\delta_{ext}(G_C) = \frac{|\gamma(G_C)|}{|S_C|.(|S|-|S_C|)}$$

The average link density of G=(S,E), denoted δ(G), is the ratio between the number of edges of G and the maximum number of possible edges of G:

$\delta(G) = |E| / ((|S|-1)/2)$

For the subgraph G_C to be a community, $\delta_{int}(G_C)$ has to be considerably larger than δ(G) and $\delta_{ext}(G_C)$ has to be much smaller than δ(G).

The edge betweenness (GIRVAN; NEWMAN, 2002) of an edge (t,u) in E is the number of pairs (w,v) of nodes in S for which (t,u) belongs to the shortest path between w and v.

Community detection algorithms search, implicitly or explicitly, for the best trade-off between a large $\delta_{int}(G_C)$ and a small $\delta_{ext}(G_C)$. They are usually classified as non-overlapping and overlapping. In non-overlapping algorithms, each node belongs to a single community. An example is the Edge Betweenness Method (EBM) (GIRVAN; NEWMAN, 2002), which finds communities by successively deleting edges with high edge betweenness. In overlapping algorithms, a node may belong to multiple communities. An example is the Greedy Clique Expansion algorithm (GCE) (LEE *et al.*, 2010), which first discovers maximum cliques to be

used as seeds of communities and then greedily expands these seeds by optimizing a fitness function. Another example is the Community Overlap Propagation Algorithm (COPRA) (GREGORY, 2010), which follows a label propagation strategy (where the labels represent the communities).

3.2.3. Clustering Validation Measures

Clustering validation measures are used to validate a clustering (or community detection) strategy against a ground truth.

Let $\boldsymbol{U}$ be the *universe*, that is, the set of all elements. Let $\boldsymbol{C} = \{C_1, C_2, ..., C_m\}$ and $\boldsymbol{T} = \{T_1, T_2, ..., T_n\}$ be two sets of subsets of $\boldsymbol{U}$.

The definitions that follow are generic, but the reader may intuitively consider $\boldsymbol{U}$ as the set of all datasets in the LOD cloud, $\boldsymbol{C}$ as a set of dataset clusters, obtained by one of the clustering algorithms, and $\boldsymbol{T}$ be a set of sets of LOD datasets taken as the ground truth (i.e., the topical domains of the LOD diagram).

Purity (MANNING et al., 2008) is a straightforward measure of cluster quality that is determined by simply dividing the number of elements of the most frequent domain contained in each cluster by the total number of elements. Purity is defined as follows:

$$purity(\boldsymbol{C}, \boldsymbol{T}) = \frac{1}{|\boldsymbol{U}|} \sum_{i=1,...,m} \max_j(|C_i \cap T_j|)$$

Purity values ranges from 0 to 1, where higher values indicate better clusters with respect to the ground truth. However, high values of purity are easy to reach when the number of clusters is large. Thus, purity is not a good trade off the quality of the clustering against the number of clusters[11].

Unlike purity, the *normalized mutual information* (NMI) (MANNING et al., 2008) offers a trade-off between the number of clusters and their quality. Intuitively, NMI is the fraction of mutual information that is contained in the

[11] https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

current clustering representation. NMI ranges from 0 to 1, where higher values indicate better clusters with respect to the ground truth, and is defined as follows:

$$NMI(\boldsymbol{C},\boldsymbol{T}) = \frac{I(\boldsymbol{C},\boldsymbol{T})}{\big(H(\boldsymbol{C}) + H(\boldsymbol{T})\big)/\,2}$$

where $I(\boldsymbol{C},\boldsymbol{T})$ represents the *mutual information* between $\boldsymbol{C}$ and $\boldsymbol{T}$ and is defined as:

$$I(\boldsymbol{C},\boldsymbol{T}) = \sum_{i=1,\ldots,m} \sum_{j=1,\ldots,n} \frac{|C_i \cap T_j|}{|\boldsymbol{U}|} \log\left(\frac{|\boldsymbol{U}|.|C_i \cap T_j|}{|C_i|.|T_j|}\right)$$

and $H(\boldsymbol{C})$ is the *entropy* of $\boldsymbol{C}$ and is defined as:

$$H(\boldsymbol{C}) = -\sum_{i=1,\ldots,m} \frac{|C_i|}{|\boldsymbol{U}|} \log\left(\frac{|C_i|}{|\boldsymbol{U}|}\right)$$

and likewise for $H(\boldsymbol{T})$, the entropy of $\boldsymbol{T}$.

The *Estimated Mutual Information* (EMI) (NUNES *et al.*, 2013) measures the dependence between $\boldsymbol{C}$ and $\boldsymbol{T}$ (intuitively, the identified clusters and the topical domains in the LOD diagram). EMI is an $m \times n$ matrix, where each element is defined as follows:

$$EMI_{i,j} = \frac{m_{i,j}}{M}.log\left(M.\frac{m_{i,j}}{\sum_{a=1}^{n} m_{i,a}.\sum_{b=1}^{m} m_{b,j}}\right)$$

where

- $[m_{i,j}]$ is the *co-occurrence matrix* of $\boldsymbol{C}$ and $\boldsymbol{T}$, with $m_{i,j} = |C_i \cap T_j|$, for $i \in [1,m]$ and $j \in [1,n]$
- $M = \sum_{i=1}^{m}\sum_{j=1}^{n} m_{i,j}$

3.2.4. Dataset Profiling Techniques

Profiling techniques address the problem of generating dataset descriptions. We will use in this paper the profiling technique described in (KAWASE *et al.*, 2014), that generates *profiles* or *fingerprints* for textual resources. The method has five steps:

1. Extract entities from a given textual resource.

2. Link the extracted entities to English Wikipedia articles.
3. Extract English Wikipedia categories for the articles.
4. Follow the path from each extracted category to its top-level category and compute a vector with scores for the top-level categories thus obtained.
5. Perform a linear aggregation in all dimensions of the vectors to generate the final profile, represented as a histogram for the 23 top-level categories of the English Wikipedia.

3.3. An Approach to Automatic Creation and Analysis of a Linked Data Cloud Diagram

The proposed process has three main steps (see Figure 1):

1. Construction of the LOD graph.
2. Dataset clusterization.
3. Dataset community description.

The first step of the process creates a graph that describes the LOD cloud, using metadata extracted from metadata catalogs (c.f. Section 3.2.1).

The second step clusters the datasets represented as nodes of the LOD graph. It applies community detection algorithms to partition the LOD graph into *communities* (also called *clusters* or *groups*) of related datasets. Intuitively, a set of datasets forms a community if there are more linksets between datasets within the community than linksets interlinking datasets of the community with datasets in rest of the LOD cloud (c.f. Section 3.2.2).

The last step generates descriptions for the dataset communities by applying a dataset profiling technique to the datasets in each community C_i identified in the previous step. As some of the datasets may contain a large number of resources, only a random sample of each dataset is considered.

Furthermore, to generate the labels that describe C_i, the profiling technique considers the literals of the datatype properties *rdfs:Label*, *skos:subject*, *skos:prefLabel* and *skos:altLabel* of the sampled resources. We recall that this

step adopts the profiling technique described in Section 3.2.4 to generate community descriptions.

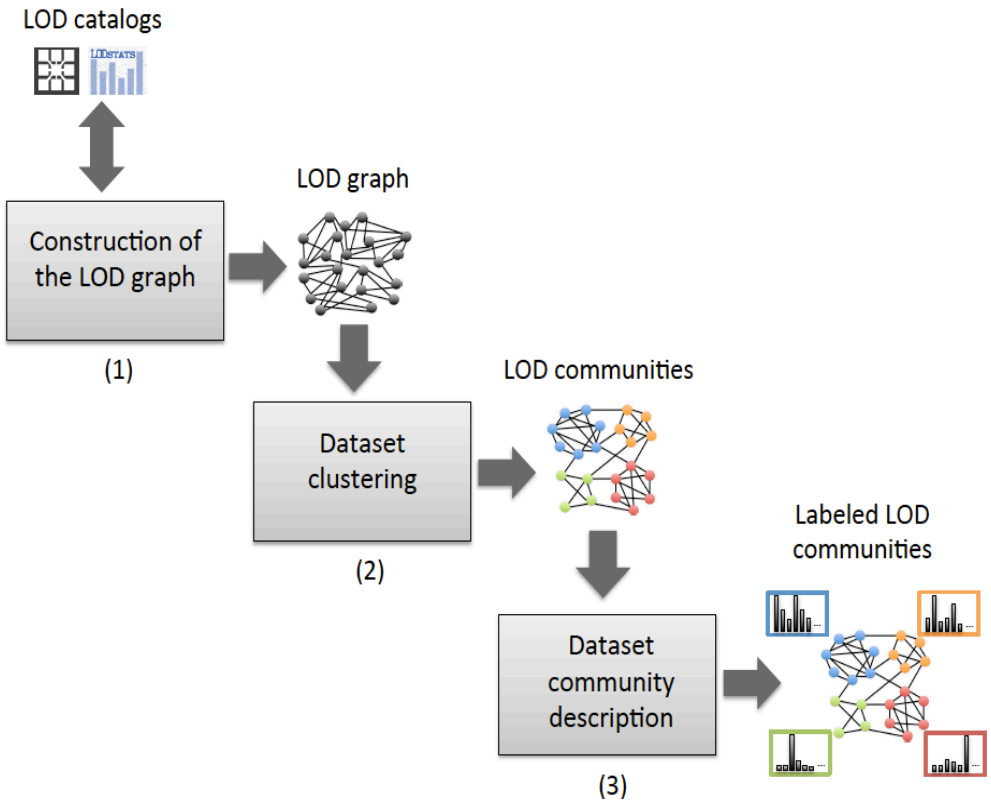


Figure 1: Community analysis process of the LOD.

3.4. Evaluation Setup

This subsection details the evaluation setup of the proposed process. Section 3.4.1 covers the construction of the LOD graph and describes the ground truth. Section 3.4.2 introduces the community detection algorithms used and discusses how the resulting communities are evaluated by taking into account the clustering validation measures described in Section 3.2.3. Finally, Section 3.4.3 analyses the labels assigned to the resulting communities, considering the expressiveness and the ability to represent the content of the datasets belonging to each community.

3.4.1.
Construction of the LOD graph and Description of the Ground Truth

To construct a LOD graph, we extracted all datasets from the Mannheim Catalog, along with their content metadata: title, description, tags and linksets. For the sake of simplicity and comparison between the ground truth and the proposed approach, we refer to the topical domains also as communities.

As ground truth, we adopted the LOD diagram described in (SCHMACHTENBERG et al., 2014) (see Section 3.2.1).

3.4.2.
Setup of the Dataset Clusterization Step

Three algorithms traditionally used in community detection and clustering problems were considered as an attempt to reproduce the LOD diagram: Greedy Click Expansion (GCE), Community Overlap PRopagation Algorithm (COPRA) and the Betweenness Method (EBM) (see Section 3.2.2). The choice of these three algorithms was based on their previously reported performance in real world scenarios (FORTUNATO, 2010; XIE et al., 2013).

We used Purity, Normalized Mutual Information (NMI) and Estimated Mutual Information (EMI) (see Section 3.2.3) as clustering validation measures. Again, these measures are estimated by comparing the results obtained with the community detection algorithms and the ground truth.

A brief description of parameterization of the three algorithms goes as follows:

- EBM: Table **3** shows the top 10 best configurations for EBM in order to reproduce the results found in the ground truth. Very briefly, the number of edges with the highest betweenness that must be removed from the LOD graph in order to detect the communities was used as stopping criterion.

Table 2: Top 10 best configurations for EBM by decreasing order of NMI.

Number of removed	**Purity**	**NMI**
600	0.60291	0.49287
550	0.60109	0.48619
300	0.56831	0.47381
650	0.54645	0.46870
700	0.51730	0.45848
500	0.60474	0.45061
750	0.49545	0.44958
450	0.58106	0.44949
800	0.46812	0.44551
850	0.39891	0.42707

Table 3: Top 10 best configurations for GCE by decreasing order of NMI.

Clique size	Overlapping rate	Alpha	Phi	**Purity**	**NMI**
3	0.0	0.8	0.2	0.4207	0.57263
3	0.0	1.0	0.8	0.3643	0.55509
3	0.0	1.0	0.2	0.3825	0.54227
3	0.1	0.8	0.6	0.4936	0.51040
3	0.0	1.2	0.2	0.4663	0.51022
3	0.1	1.2	0.2	0.4881	0.50926
3	0.0	0.8	0.4	0.3442	0.50534
3	0.0	1.2	0.8	0.5082	0.50148
3	0.2	1.0	0.2	0.5664	0.49747
3	0.3	0.8	0.2	0.4845	0.49542

- GCE: Table **3** shows the top 10 best configurations for GCE in order to reproduce the results found in the ground truth. Very briefly, the *Alpha* and *Phi* parameters were used to control the greedy expansion and to avoid duplicate cliques/communities, respectively.
- COPRA: Table **4** shows the best configuration for COPRA. As COPRA is nondeterministic, the tuning of its parameters was obtained by the average of 5-cycle runs.

Unlike EBM, GCE and COPRA are capable of finding overlapping communities. However, as the ground truth defines non-overlapping communities, these algorithms obtained the best results when the overlapping rate/parameter was set to 0 (no overlap between datasets) and 1 (one label per dataset), respectively.

Table 4: Best quality results for the community detection/clustering algorithms.

Algorithm	#Clusters	Purity	NMI
GCE	6	0.42	0.57
COPRA	4	0.30	0.32
EBM	18	0.60	0.49

3.4.3. Setup of the Dataset Community Description Step

Although the Mannheim Catalog lists 1,014 datasets, only a fraction of the listed datasets has SPARQL endpoints available. At the time of this evaluation, approximately 56% of the SPARQL endpoints were up and running. For each available dataset, a sample of 10% of its resources were extracted and used as input to the *fingerprints* algorithm (see Sections 3.2.4 and 3.3), which assigned labels to the communities automatically generated by the best performing parameterization of the GCE algorithm.

3.5. Results

The first part of the discussion addresses the performance of the dataset clusterization step. The second part presents the results for the dataset community description step.

3.5.1. Performance of the Dataset Clusterization Step

Quality of the generated communities. As shown in Table 4, GCE obtained the highest NMI value, 0.57, and EBM the highest purity value, 0.60. The high NMI value achieved by GCE indicates a mutual dependence between the communities found by the algorithm and those described in the ground truth. Despite the highest purity value obtained by EBM, this technique was not consistent with the communities in the ground truth. COPRA obtained low values for both purity and NMI, indicating that the resulting communities and those induced by the ground truth do not match.

Communities detected. Table **5** and Table **6** show the co-occurrence and estimated mutual information matrices, respectively, for the best performing parameterization of the GCE algorithm. The first column shows the communities (domains) of the ground truth, whereas columns labeled 0-5 represent the communities found by GCE.

Table 5: Co-occurrence matrix of the GCE result.

Domain/Community	**0**	**1**	**2**	**3**	**4**	**5**
Social Networking	0	**88**	0	0	0	0
User-generated Content	0	4	0	0	0	0
Geographic	0	2	4	0	0	0
Publications	**37**	4	1	1	0	0
Cross-Domain	1	2	0	0	0	0
Life Sciences	0	2	0	**13**	**24**	0
Government	1	1	**10**	1	0	**59**
Media	0	2	0	1	0	0

The light gray cells in Table 6 mark the highest dependencies between the topical domains extracted from the ground truth and the communities generated by GCE. Note that, due to the low level of dependency between the ground truth categories

"Cross-Domain", "Media" and "User-Generated Content" (UGC) and the clusters found by GCE, datasets in these ground truth categories communities are possibly split over several clusters.

Table 6: EMI matrix of the GCE result.

Domain/Community	0	1	2	3	4	5
Social Networking	0	0.262	0	0	0	0
User-generated Content	0	-0.005	0	0	0	0
Geographic	0	-0.003	0.013	0	0	0
Publications	0.092	-0.013	-0.002	-0.002	0	0
Cross-Domain	-0.002	-0.005	0	0	0	0
Life Sciences	0	-0.007	0	0.046	0.095	0
Government	-0.004	-0.006	0.018	-0.003	0	0.150
Media	0	-0.003	0	0.001	0	0

3.5.2. Performance of the Dataset Community Description Step

Table **7** shows the labels generated by the dataset community description method adopted (see Section 3.2.4). These labels were assigned to the communities found by the best performing parameterization of the GCE algorithm. The first column shows the 23 top-level categories of Wikipedia, whereas columns labeled 0-5 represent the communities found by GCE. To facilitate a comparison between the labels in different communities, we normalized the scores assigning 1.0 to the category with the highest score. The light gray cells mark the strongest relations between the categories from the generated labels and the communities generated by GCE. We recall that Table 1 presents the manually assigned labels given to the communities in the ground truth.

Table 7: Histograms of top-level categories for each community structure.

Category / Community	0	1	2	3	4	5
Agriculture	0	0	0.39	0.03	0.02	0.03
Applied Science	0.80	0.34	0.37	0.06	0.11	0.03
Arts	0.03	0.11	0	0	0.01	0.03
Belief	0.03	0.04	0	0	0	0.02
Business	0.59	0.53	0.11	0.03	0.03	0.27
Chronology	0.04	0.15	0.02	0.01	0	0.06
Culture	0.13	0.19	0.27	0	0.03	0.11
Education	0.20	0.06	0.06	0.04	0.12	0.08
Environment	0.01	0.03	0.40	0.02	0.02	0.10
Geography	0.05	0.11	1.00	0.13	0.03	0.70
Health	0.05	0.06	0.41	0.18	0.65	0.03
History	0.06	0.03	0.11	0.06	0.02	0.13
Humanities	0.04	0.08	0	0	0.2	0
Language	0.20	0.10	0.01	0	0.02	0.03
Law	0.04	0.45	0.10	0.01	0.02	0.24
Life	0.08	0.03	0.96	1.00	1.00	0.02
Mathematics	0.60	0.03	0.03	0.03	0.03	0.02
Nature	0.29	0.08	0.24	0.03	0.06	0.03
People	0.02	0.52	0.02	0.01	0.03	1.00
Politics	0.05	0.35	0.12	0.03	0.01	0.65
Science	1.00	0.16	0.26	0.03	0.10	0.03
Society	0.32	1.00	0.14	0.06	0.05	0.32
Technology	0.61	0.37	0.11	0.01	0.02	0.08

3.6. Discussion and Analysis

3.6.1. An Analysis of the Dataset Clusterization Results

This subsection analyses the dataset clusterization results. The analysis compares the dataset communities found in the clustering step – referred to as *Community 0* to *Community 5* – with the dataset topical domains defined in the LOD diagram (SCHMACHTENBERG et al., 2014) – "Media", "Government", "Publications", "Geographic", "Cross-Domain", "Life Sciences", "Social Networking" and "User-generated content" – taken as ground truth.

As shown in Section 3.5, the GCE algorithm did not recognize as communities the datasets classified in the "Cross-domain", "Media" and "User-Generated Content" (UGC) domains. A possible reason for the lack of a cross-domain community lies in its own nature, that is, cross-domain datasets tend to be linked to datasets from multiple domains, acting as hubs for different communities. Another (interesting) reason is that cross-domain datasets do not contain a large number of links between themselves. The lack of links between cross-domain datasets results in a subgraph with low density, which GCE does not consider a new community. Nevertheless, if overlapping rates are considered, datasets that belong to several communities may generate a cross-domain community. Likewise, the "Media" community presented a low density due to its low number of linksets.

Community 0 presents a high concentration of datasets from the "Publications" domain, including datasets from the ReSIST project[12], such as `rkb-explorer-acm`, `rkb-explorer-newcastle`, `rkb-explorer-pisa` and `rkb-explorer-budapest`. This led us to assume that this community is equivalent to the "Publications" domain.

Community 1 is the largest community among those recognized and contains mostly datasets from the "Social Networking" domain. This community

[12] `http://www.rkbexplorer.com/`

includes datasets such as `statusnet-postblue-info`, `statusnet-fragdev-com`, `statusnet-bka-li` and `statusnet-skilledtestes-com`.

Contrasting with the previous communities, *Community 2* includes datasets from two different domains, "Government" and "Geographic". Note that datasets in these two domains share a considerable number of linksets, which led GCE to consider them in the same community. Government datasets often provide statistical data about places, which may justify such a large number of linksets between them. *Community 2* includes datasets from the "Government" domain, such as `eurovoc-in-skos`, `gemet`, `umthes`, `eea`, `eea-rod`, `eurostat-rdf` and `fu-berlin-eurostat`. It also includes datasets from the "Geographic" domain, such as `environmental-applications-reference-thesaurus` and `gadm-geovocab`.

Communities 3 and *4* are equivalent to only one domain, "Life Sciences". Intuitively, the original "Life Sciences" domain was split into *Community 3*, containing datasets such as `uniprot`, `bio2rdf-biomodels`, `bio2rdf-chembl` and `bio2rdf-reactome`, and into *Community 4*, containing datasets such as `pub-med-central`, `bio2rdf-omim` and `bio2rdf-mesh`. A distinction between these two communities becomes apparent by inspecting the datasets content: *Community 3* is better related to Human Biology data (about molecular and cellular biology), whereas *Community 4* is better related to Medicine data (about diagnosis and treatment of diseases).

Finally, *Community 5* groups datasets from the "Government" domain. Examples of datasets in this community are `statictics-data-gov-uk`, `reference-data-gov-uk`, `opendatacommunities-imd-rank-2010` and `opendatascotland-simd-education-rank`.

3.6.2. An Analysis of the Dataset Community Description Results

This subsection analyses the dataset community description results (see Table 6). For each dataset community, the analysis compares the 23-dimension vector description automatically assigned by the fingerprint approach with the labels manually assigned by the ground truth. In what follows, we say that a vector v has a peak for dimension i iff $v_i \geq 0.50$.

Community 0, which is equivalent to the "Publications" domain, is described by a vector with peaks for "Applied Science", "Business", "Mathematics", "Science" and "Technology". The presence of five categories shows the diversity of the data in this community. We consider that the label "Publications" assigned by the ground truth classification is better related to the tasks developed in this community than the semantics of the data itself. The rationale behind this argument is that the data come from scholarly articles published in journals and conferences.

Community 1, which is equivalent to the "Social Networking" domain, is described by a vector with peaks for "Business", "People" and "Society". Clearly, the vector was able to capture the essence of social data, covering topics related to the society in general.

Community 2, which has datasets from two different domains, "Government" and "Geographic", is described by a vector with peaks for "Geography" and "Life". Geographic data are available in various domains and, for this reason, the data cannot be described by a single category.

Community 3, which is partially equivalent to the "Life Sciences" domain, is described by a vector with a single peak for "Life", which is similar to the manually assigned domain. *Community 3* is complemented by *Community 4*, whose vector has peaks for "Health" and "Life". Taking into account these two vectors, we may identify datasets in this community with two different content profiles.

Community 5, which is equivalent to the "Government" domain, is described by a vector with peaks for "Geographic", "People" and "Politics". The vector also has significant values for "Business", "Law" and "Society". In general, datasets in this community are related to government transparency. For this reason, the vector for *Community 5* shows an interesting presence of "People", "Society" and "Politics".

3.7. Conclusion

This chapter presented a novel, automatic analysis of the Linked Open Data cloud through community detection algorithms and profiling techniques. The results indicate that the best performing community detection algorithm is the GCE algorithm, with NMI and purity values of 0.57 and 0.42, respectively. Although the EBM algorithm obtained the highest purity value, the high number of communities led to a low NMI value. The mutual dependence between the communities generated using GCE and those from the ground truth is also not high, but, as discussed in Section 3.6, the lack of linksets between datasets in some domains, such as "Cross-Domain", implies a need for the re-organization of datasets as well as the merging and splitting of communities.

The next part of the evaluation focused on comparing the labels manually assigned by the ground truth with the description automatically generated by the profiling technique. It is important to highlight that the labels in the ground truth were created based on classification criteria such as the nature of the data and the actions developed by the organization or institution that produces the data. For example, most datasets labeled as "Publications" provides information about computer science scientific articles. However, this label is better related to the fact that articles need to be published to be part of these databases, instead the content itself. By contrast, the automatic process relied on the contents of the datasets to generate the community labels.

The experimental results showed that the proposed process automatically creates a clusterization of the LOD datasets which is consistent with the traditional LOD diagram and that it generates meaningful descriptions of the dataset communities. Moreover, the process may be applied to automatically update the LOD diagram to include new datasets.

To conclude, we note that, in the first step of the approach proposed in this chapter, we used only linksets to build the LOD graph, that is, no dataset content was taken into account. In the next chapter, we will present an approach based on a clustering strategy that considers dataset content to obtain datasets clusters.

For additional information, including graphical visualizations and detailed results, we refer the reader to the Web site available at `http://drx.inf.puc-rio.br:8181/Approach/communities.jsp`.

4
Dataset interlinking recommendation

This chapter describes three different approaches to face the problem of dataset interlinking recommendation. The first approach uses link prediction measures and was implemented in the TRT tool. The second approach utilizes machine-learning algorithms and was implemented in the TRTML tool. The last approach uses profiling and clustering techniques and was implemented in the DRX tool.

4.1.
Introduction

Despite the efforts to foster publishing data as Linked Open Data (LOD) (BIZER, 2011), data publishers still face difficulties to integrate their data with other datasets available on the Web (BIZER et al., 2009). However, defining RDF links between datasets helps improve data quality, allowing the exploration and consumption of the existing data.

We may divide the question of defining RDF links between two datasets into two problems. We refer to the problem of creating RDF links between a source dataset and a target dataset as the *dataset interlinking problem* and to the problem of recommending target datasets to be interlinked with a given source dataset as the *dataset interlinking recommendation problem*.

This chapter introduces three approaches to deal with the dataset interlinking recommendation problem. Each approach can be used depending of the data or metadata that data publishers have at hand. Suppose a data publisher wants to obtain dataset interlinking recommendations for a given dataset *t*. If he has some intuition about which datasets can be interlinked with *t*, that is, if he can define a *target-context* for *t*, he may use the TRT tool, which needs as an input the target-context or *t* and a selected link-prediction measure. If the data publisher has a VoID file with the definition of vocabularies, classes and properties employed in *t*, he may use the TRTML tool. If *t* has a SPARQL endpoint, the data publisher

may use the DRX tool. It is important to mention that these tools can be used in complementary ways. For example, the data publisher may obtain the target context required for TRT from the dataset interlinking recommendations returned by either TRTML or DRX.

The remainder of this chapter is structure as follows. Sections 4.2, 4.3 and 4.4 respectively describe the TRT, TRTML and DRX tools and their underlying approaches. Section 4.5 presents an evaluation and a comparison of the tools. Finally, Section 4.6 presents the conclusions of this chapter.

We note that the experiments presented in Sections 4.2, 4.3 and 4.4 were carried out to discover the best configuration of TRT, TRTML and DRX tools. Section 4.5 contains a complete evaluation of the tools, which includes a comparison of their features and their performances.

4.2. TRT- The Dataset Recommendation Tool

4.2.1. An Approach to Dataset Interlinking Recommendation

Recall that a dataset *t* is a set of RDF triples. A resource, identified by an RDF URI reference *s*, is defined in *t* iff *s* occurs as the subject of a triple in *t*.

Let *t* and *u* be two datasets. A *link* from *t* to *u* is a triple of the form (*s, p, o*), where *s* is an RDF URI reference identifying a resource defined in *t* and *o* is an RDF URI reference identifying a resource defined in *u*; we also say that (*s, p, o*), *interlinks s* and *o*. We say that *t can be interlinked with u* iff it is possible to define links from *t* to *u*.

The recommendation procedure analyses the Linked Data network in much the same way as a Social Network. Therefore, our graph model is defined as follows. A *Linked Data network* is a graph $G = (S, C)$ such that *S* is a set of datasets and *C* contains an edge (*t, u*), called a connection from *t* to *u*, iff there is at least one link from *t* to *u*.

In order to identify the strong connections among datasets in the LD network the procedure uses link prediction theory, which estimates the likelihood

of the existence of a link between datasets. Our approach focuses on local and quasi-local indices to measure the structural similarity between datasets (LÜ *et al.*, 2009) according to their link structure (see Figure **2**).

Indice		Equation
Type	**Name**	
Local indices	Common Neighbors	$CN_{t,u} = \|C_t \cap C_u\|$
	Salton	$Salton_{t,u} = \frac{\|C_t \cap C_u\|}{\sqrt{C'_t . C'_u}}$
	Jaccard	$Jaccard_{t,u} = \frac{\|C_t \cap C_u\|}{\|C_t \cup C_u\|}$
	Sørensen	$Sørensen_{t,u} = \frac{2.\|C_t \cap C_u\|}{C'_t + C'_u}$
	Hub Promoted index	$HPI_{t,u} = \frac{\|C_t \cap C_u\|}{min\{C'_t, C'_u\}}$
	Hub Depressed index	$HDI_{t,u} = \frac{\|C_t \cap C_u\|}{max\{C'_t, C'_u\}}$
	Leicht-Holme-Newman	$LHN_{t,u} = \frac{\|C_t \cap C_u\|}{C'_t . C'_u}$
	Preferencial Attachment	$PA_{t,u} = C'_t . C'_u$
	Adamic-Adar	$AA_{t,u} = \sum_{w \in C_t \cap C_u} \frac{1}{log \|C'_w\|}$
	Resource Allocation	$RA_{t,u} = \sum_{w \in C_t \cap C_u} \frac{1}{\|C'_w\|}$
Quasi-local indices	Local Path	$LP_{t,u} = A^2 + \varepsilon A^3$
	Local Random Walk	$LRW_{t,u}(s) = \frac{C'_t}{2\|C\|} . \pi_{t,u}(s) + \frac{C'_u}{2\|C\|} . \pi_{u,t}(s)$

Figure 2: Local and quasi-local indices.

where:

- C_{di} is the context of d_i (datasets that d_i points to), where d_i a specific dataset;
- C'_{di} is the inverse context of d_i (datasets that point to d_i), where d_i a specific dataset;
- A_j is the number of different paths with length j connecting t and u;
- ε is a free parameter;
- $\pi_{t,u}(s)$ is the probability that a random walker starting on t locates u after s steps;
- C is the set of all edges of the Linked Data network G.

In what follows, we describe the Common Neighbours (CN) index in order to understand how the indices work in practice. Thus, given a dataset t, let C_t denote the set of neighbors of t. Intuitively, two datasets, t and u, are more likely to have a connection if they have many common neighbors. This probability can

be estimated by counting the overlapping of the neighbors or by counting the number of different paths with length 2 connecting *t* and *u* (LÜ *et al.*, 2009).

The dataset interlinking recommendation procedure starts by building the Linked Data network $G = (S, C)$. Then, given a target dataset *t* not in *S* (intuitively the user wishes to define links from *t* to the datasets in *S*) and a target context C_t for *t* consisting of one or more datasets *u* in *S* (intuitively the user knows that *t* can be interlinked with *u*), the procedure employs one of the local or quasi-local indices to measure the likelihood of dataset *t* being connected with datasets in *S*. Finally, the procedure uses these probability scores to produce an order list *L* of datasets in *S*, called a ranking.

4.2.2. TRT Architecture

The TRT tool has two modules and a database, depicted in Figure 3, which are distributed in three different layers: data acquisition, data processing and application. These modules perform two main tasks:

1. Collect metadata features from datasets in the LOD cloud.
2. Provide dataset recommendations based in link prediction measures.

The data acquisition layer includes the crawling module, which discovers metadata about the LOD datasets from LOD catalogs. Since link prediction measures use linksets to estimate the likelihood of the existence of a link between datasets, only LOD datasets with this feature are considered. The crawling module uses the CKAN API[13] to query metadata available in such catalogs.

[13] http://ckan.org/

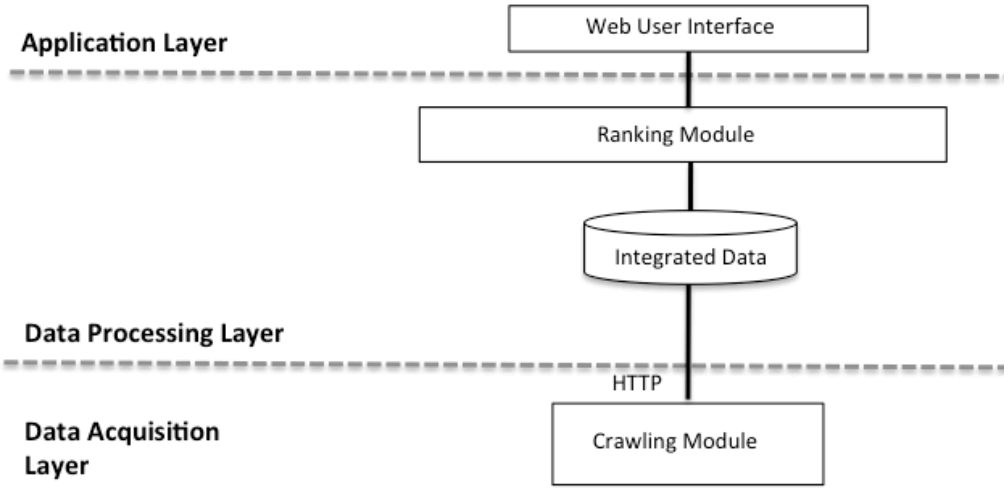


Figure 3: Architecture TRT tool.

The data processing layer includes a database and a module: the *integrated data* database stores the metadata retrieved from the LOD catalogs. Finally, the ranking module implements the link prediction measures presented in Figure **2** in order to provide dataset interlinking recommendations for a given dataset.

4.2.3. TRT GUI

Briefly, suppose that the user is working on a dataset *t* and wants to discover one or more datasets *u* such that *t* can be interlinked with *u*. He then uses the tool to obtain recommendations. The tool first builds the Linked Data network $G = (S, C)$ defined by the metadata stored in any metadata repository that offers the CKAN API. Then, the user defines the rest of the input data the tool requires. He may define a target context C_t for *t*, consisting of one or more datasets in S, in two different ways: (i) by providing a VoID descriptor V_t for *t* from which the tool extracts C_t by analysing the *void:linkset* declarations occurring in V_t; or (ii) by manually selecting datasets from the categories the tool displays. Finally, the user chooses a similarity index from those shown on Figure **2**. From this input data, the tool outputs a ranked list of datasets, thereby helping reduce the effort required to

find related datasets for the interlinking process. Figure **4** depicts the user interface of the TRT tool.

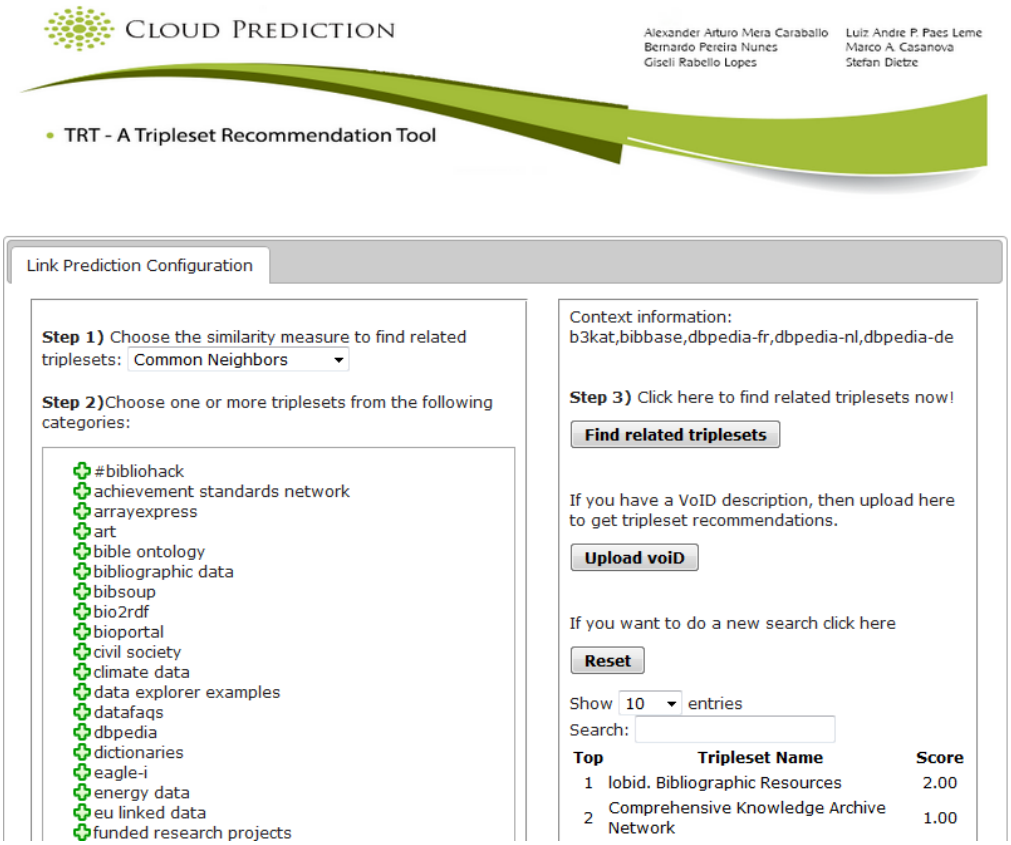


Figure 4: TRT tool interface.

4.2.4. Evaluation Setup

The tool was evaluated using the DataHub repository, which contains more than 6,000 datasets, with approximately 15 thousand links that connect only 711 of the available datasets. The links across datasets were used to rank and recommend datasets for interlinking. The recommendation process was assessed using the 10-fold cross validation approach, as in (LOPES *et al.*, 2013), where they split the Linked Data graph $G = (S,C)$ into *recommendation partitions* and *testing partitions* in ten different ways, and defined *target context* as follows:

- Given $G = (S,C)$, a *recommendation partition* is a subgraph $G_i = (S_i,C_i)$ such that S_i is a set of datasets to be considered for

recommendation and C_i is the set of links among the datasets in S_i provided by the linksets in the catalogs

- A *testing partition* is a pair $Tp_i = (T_i, aC_i)$ such that T_i is the set of dataset in S, but not in S_i, called *recommendation targets*, and aC_i is a set of sets such that, for each $t \in T_i$, aC_i contains the set aC_t of all datasets u in S_i such that there is a connection from t to u in C
- For each recommendation target $t \in T_i$, a *target context* C_t consists of some chosen datasets in aC_t.

Additionally, for each different recommendation partition $G_i = (S_i, C_i)$, testing partition $Tp_i = (T_i, aC_i)$, recommendation target $t \in T_i$, with target context $C_t \in aC_i$, they defined:

- the *gold standard* for t is defined as the set $Gs_t = aC_t - C_t$ and represents the datasets that must be recommended
- a *relevant datasets* to be recommended for t is a dataset in Gs_t
- a *candidate dataset* to be recommended for t is a dataset in $S_i - C_t$

We consider in the experiments traditional Information Retrieval measures such as Recall and Mean Average Precision (MAP).

The *overall Recall* is the mean of the recall of each testing partition. The recall of a testing partition Tp_i is defined as the average of the recall values of each dataset $t_j \in T_i$:

$$Recall(Tp_i) = \frac{\sum_{j=1}^{|T_i|} Recall(t_j)}{|T_i|}$$

where:

- $Recall(t_j)$ is defined as the proportion between the number of relevant datasets that are recommended for t_j and the total number of datasets that must be recommended $|Gs_{tj}|$.

The *overall MAP* is defined as the mean of the MAP of each testing partition. The MAP of a testing partition Tp_i is in turn defined as the mean of the average precision scores of each dataset $t_j \in T_i$:

$$MAP(Tp_i) = \frac{\sum_{j=1}^{|T_i|} AveP(t_j)}{|T_i|}$$

where $AveP(t_j)$ is the average precision in the ranking of the dataset t_j. It is computed as an average of the precision values obtained for each relevant dataset.

4.2.5. Results

Figure **5** summarizes the results for different target context sizes (shown in the first column of the table). The entries corresponding to the highest results among the 12 indices are emphasized in boldface underlined.

The reader may observe that the PA index obtained the best MAP (37.83%) for target contexts with very few datasets, while the RA index turned out to be more precise (72.42%) for larger target contexts. Table 2 also shows the coverage results. The PA index obtained the highest recall (96.4%), regardless of the size of the target context.

MAP	CN	Salton	Jaccard	Sørensen	HPI	HDI	LHN	PA	AA	RA	LP	LRW
1	18.17	14.49	16.30	14.73	17.08	15.00	14.80	**37.83**	18.06	17.80	18.46	15.57
5	49.48	25.07	21.80	20.36	35.14	19.20	18.38	48.26	**52.20**	51.48	58.23	26.05
10	63.49	30.99	30.40	28.71	41.81	24.41	19.44	52.62	63.43	**63.71**	62.63	31.91
20	71.20	34.22	44.37	43.56	38.14	34.14	17.90	53.97	71.46	**72.38**	70.59	34.66
50	71.13	27.73	69.49	70.55	20.64	66.14	15.92	47.30	70.99	**72.42**	67.51	39.03
Recall	CN	Salton	Jaccard	Sørensen	HPI	HDI	LHN	PA	AA	RA	LP	LRW
1	48.72	49.69	49.86	49.76	49.68	49.55	50.02	**96.40**	50.81	48.74	48.81	49.12
5	81.45	83.80	82.69	83.03	83.68	82.23	82.43	**98.45**	83.63	82.83	86.90	82.42
10	89.52	88.73	89.42	89.29	89.35	89.85	89.17	**98.74**	89.49	89.28	88.96	89.21
20	90.03	90.31	89.68	89.18	89.12	89.53	89.19	**99.80**	89.50	89.58	89.84	90.01
50	90.05	90.16	90.15	90.21	90.04	89.38	88.45	**99.56**	89.06	89.58	89.02	89.71

Figure 5: MAP and Recall of the local and quasi-local indices.

4.2.6. Conclusions

In this section, we proposed the use of link prediction techniques to address the dataset interlinking recommendation problem in the Linked Data domain and presented the TRT tool that implements the techniques. The tool computes local and quasi-local indices to predict links between datasets. The results showed that the tool performs better, with respect to recall, when the PA index is adopted. In terms of MAP, the PA index should be adopted for smaller context sizes, while the RA index should be adopted for larger context sizes.

4.3. TRTML - A Dataset Recommendation Tool based on Supervised Learning Algorithms

4.3.1. An Approach to Dataset Interlinking Recommendation

Let $D = \{d1, ..., dn\}$ be a set of datasets considered in the recommendation process and t be the dataset one wants to receive recommendations for interlinking. Instead of providing a restricted list of recommendations, we define the task of recommending datasets to be interlinked with t as a task of ranking datasets d_i in D according to the estimated probability that one can define links between resources of t and d_i. To generate the rankings, we explore an approach that combines link prediction measures and machine learning techniques.

The approach uses link prediction measures to estimate the likelihood of the existence of a link between datasets. To estimate the measures, we construct a bipartite graph $G = (D, F, E)$ consisting of two disjoints sets of nodes representing datasets D and features F. The set of edges E represents the association between the datasets and their features. The set of features of a dataset t, F_t, correspond to the vocabularies, classes or properties extracted from the VoID descriptions defined in t. The tool implements four of the traditional link prediction measures, summarized in Figure **6**, where:

- F_{di} is the feature set of dataset d_i (direct neighbors of d_i in G);
- D_{fj} is the set of datasets having feature f_j (direct neighbors of f_j in G).

Measure	Equation
Common Neighbors	$CN_{t,d_i} = \lvert F_t \cap F_{d_i} \rvert$
Jaccard coefficient	$Jaccard_{t,d_i} = \frac{\lvert F_t \cap F_{d_i} \rvert}{\lvert F_t \cup F_{d_i} \rvert}$
Preferential Attachment	$PA_{t,d_i} = \lvert F_t \rvert . \lvert F_{d_i} \rvert$
Resource Allocation	$RA_{t,d_i} = \sum_{f_j \in F_t \cap F_{d_i}} \frac{1}{\lvert D_{f_j} \rvert}$

Figure 6: Link prediction measures.

These link prediction measures were selected since they demonstrated good performance in previous work (CARABALLO *et al.*, 2013; LOPES *et al.*, 2013).

The approach uses supervised learning algorithms to learn if a pair of datasets can be interlinked, using as training set the existing links between datasets. Specifically, we build a J48 decision tree (Quinlan's C4.5 implementation), where the nodes represent the measures reported in Figure **6**, estimated using different feature sets (vocabularies, classes or properties). The leaf nodes represent the values of a binary class such that, given two datasets (t, d_i), 1 represents that d_i can be recommended to t and 0 denotes that d_i is not a good candidate to be recommended to t.

4.3.2. TRTML Architecture

The TRTML tool is based on three modules, depicted in Figure 7, which are distributed in three different layers: data acquisition, data processing and application. These modules perform five main tasks:

1. Parse VoID descriptor to extract vocabularies, classes and properties.
2. Process the extracted metadata features and update the predictive model.
3. Provide dataset recommendations.

The data acquisition layer includes the parsing module, which retrieves metadata features from manually submitted VoID descriptors. This module implements the Jena[14] API, an open source Semantic Web Framework for JAVA[15].

[14] https://jena.apache.org/

[15] https://www.java.com

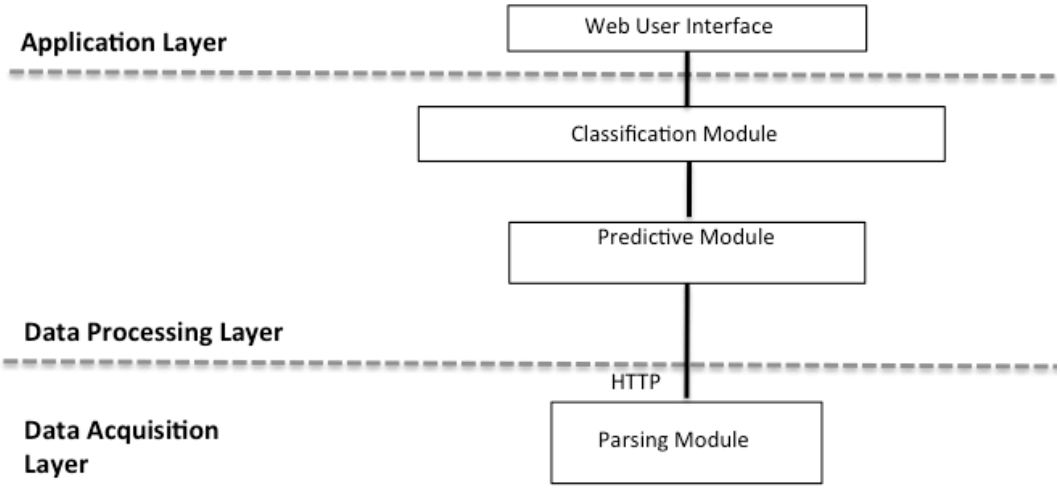


Figure 7: Architecture TRTML tool.

The data application layer includes two modules: the predictive module that uses the retrieved metadata features to recreate the predictive module. This module uses the WEKA[16] JAVA API in order to build supervised predictive models. Finally, the classification module provides a list of datasets label with one of the following classes: 0 means low probability to have common resources, 1 means high probability to have overlapped resources.

4.3.3. TRTML GUI

Suppose that a user is working on a dataset *t* and that he wants to discover one or more datasets d_i such that *t* can be interlinked with d_i. He then uses the tool to obtain dataset recommendations. First, the tool builds a classifier over the set of VoID descriptions, obtained from the DataHub catalog.

Then, the user defines the rest of the input data the tool requires: (i) he selects the serialization format of the VoID descriptor (TURTLE, RDF/XML or N-TRIPLE N3); and (ii) uploads a VoID descriptor V_t for t from which the tool extracts the feature set F_t by analyzing the *void:vocabulary, void:class* and

[16] https://weka.wikispaces.com/Use+WEKA+in+your+Java+code

void:property occurring in V_t. Finally, the tool applies the classifier, using F_t, and outputs a list of datasets that can be classified in one of the two classes (1/0): datasets label with 1 means that they have high probability to have overlapped resources, and datasets label with 0 means that they have low chance to have overlapped resources.

4.3.4. Evaluation Setup

4.3.4.1. Dataset

For this evaluation, we use the VoID descriptions stored in the DataHub catalog. We obtained a set *D* of 293 datasets whose VoID descriptions indicated the vocabularies, classes and properties the dataset used. Out of the 42,778 possible links, we uncovered a set *L* of 410 links connecting such datasets by analyzing the *void:linkset property*.

4.3.4.2. Ground truth.

Due to the lack of benchmarks for validating the creation of links between datasets, we adopted as ground truth the set *L* of links defined above. Furthermore, we separated the dataset pairs in $D \times D$ into two classes: (i) (ground truth) linked dataset pairs that are connected by a link in L, and (ii) (ground truth) unlinked dataset pairs that are not connected by a link in *L*.

4.3.4.3. Evaluation Metrics

To validate the recommendation algorithms, we adopted the standard metrics Recall *(R)*, Precision *(P)* and F-measure *(F1)*, defined based on true positives *(TP)*, true negatives *(TN)*, false positives *(FP)* and false negatives *(FN)* links between datasets. Briefly, the positive and negative terms refer to link prediction, while true and false refer to the links in *L*.

Thus, precision is defined as:

$$P = \frac{TP}{TP + FP}$$

where TP is the number of dataset correctly recommended and FP is the number of dataset wrongly recommended that are not in L.

As for the recall, it is defined as follows:

$$R = \frac{TP}{TP + FN}$$

where, *FN* indicates the missed dataset recommendations.

Finally, F1 measure the harmonic average between precision and recall.

$$F1 = 2 * \frac{P * R}{P + R}$$

4.3.4.4. Learning Algorithms

In our experiments, we additionally used two standard supervised learning algorithms – Support Vector Machines (SVM) and Multilayer Perceptron – to classify pairs of datasets into (ground truth) linked dataset pairs and (ground truth) unlinked dataset pairs, based on link prediction measures values estimated considering different features sets.

Briefly, we used the implementation of SVM provided in the Library for Large Linear Classification (LibLINER), which is an open source library for large-scale linear classification. It supports logistic regression and linear support vector machines. In the case of Multilayer Perceptron, we used a neural network that is trained using back propagation algorithm, capable of expressing a rich variety of nonlinear decision surfaces.

4.3.4.5. Results

Before discussing the results, we observe that a pair of datasets may not be in *L*, the set of links obtained from the DataHub catalog, because of a lack of

metadata information or because they were never interlinked, but they might be. This indeterminacy might contaminate the learning algorithms.

Therefore, and aiming at balancing the dataset pairs with different classes in the ground truth, we decided to vary the percentage of (ground truth) unlinked dataset pairs considered when analyzing the performance of the various algorithms. Figure **8**, Figure **9** and Figure **10** show the precision, recall and F-measure, respectively, achieved when the percentage of (ground truth) unlinked dataset pairs varies (100%, 75%, 50%, 25% and 1%), while maintaining the number of (ground truth) linked dataset pairs constant.

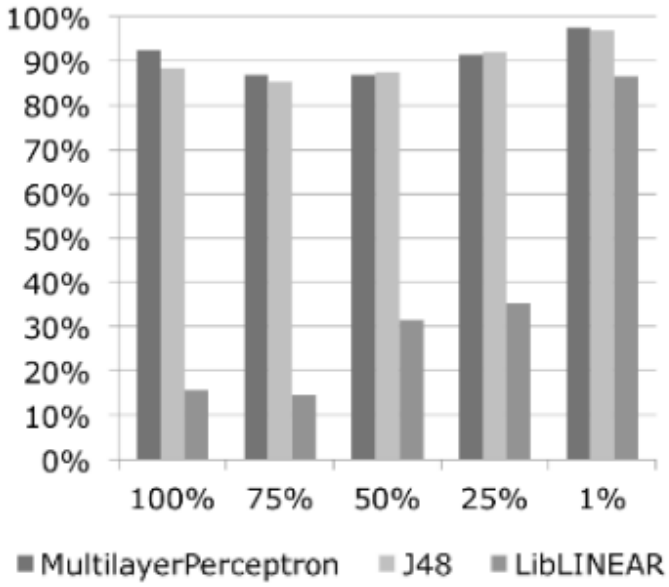


Figure 8: Precision of the supervised classifiers by the percentage of (ground truth) unlinked dataset pairs considered (100%, 75%, 50%, 25% and 1%)**.**

Figure **8** shows that both the Multilayer Perceptron and the J48 implementations achieved a precision greater than 85%, independently of the percentage of (ground truth) unlinked dataset pairs considered. Figure **9** indicates that the recall of the supervised classifiers increases when the percentage of (ground truth) unlinked dataset pairs is reduced. Figure **10** shows that the J48 algorithm obtained the best overall performance, independently of the percentage of (ground truth) unlinked dataset pairs considered.

To conclude, the J48 implementation achieved higher recall and F-measure, independently of the percentage of (ground truth) unlinked dataset pairs considered.

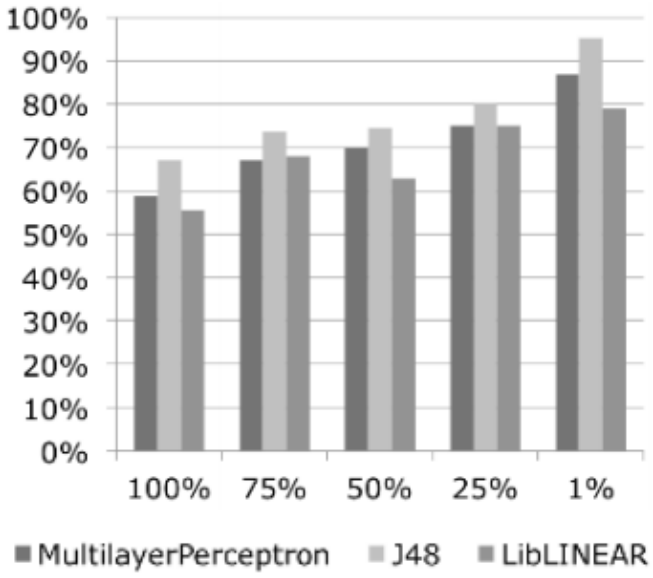


Figure 9: Recall of the supervised classifiers by the percentage of (ground truth) unlinked dataset pairs considered (100%, 75%, 50%, 25% and 1%).

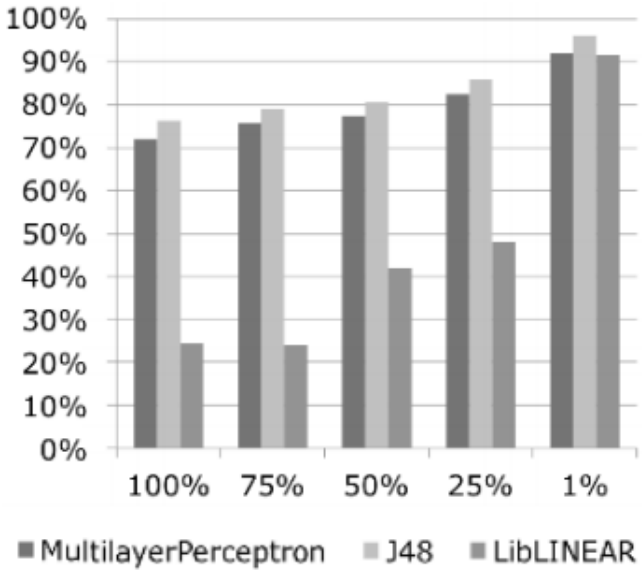


Figure 10: F-measure of the supervised classifiers by the percentage of (ground truth) unlinked dataset pairs considered (100%, 75%, 50%, 25% and 1%).

4.3.5. Conclusions

In this section, we presented a tool for *dataset interlinking recommendation*, called TRTML, which reduces the effort of searching for related datasets in large data repositories. TRTML is based on link prediction measures and supervised learning algorithms. The crucial role of the supervised learning algorithms is to automatically select a set of features, extracted from the VoID vocabulary, and a set of link prediction measures that, when combined, lead to effective dataset interlinking recommendations. After a comprehensive evaluation of the supervised learning algorithms, the results show that the implementation based on the J48 decision tree (Quinlan's C4.5 implementation) achieved the best overall performance, when compared with the Multilayer Perceptron and the SVM algorithms.

4.4. DRX - A LOD Dataset Interlinking Recommendation Tool

In this section, we present an approach implemented in the DRX tool, designed to address the dataset interlinking recommendation problem. Briefly, the approach proceeds as follows. The first step collects data from datasets on the LOD cloud; the second step creates dataset profiles. The approach considers to store these profiles for later use. When a data publisher wants to interlink a source dataset *t* with other datasets, the third step applies the same profiling technique to *t* and finally the last step outputs an ordered list of datasets whose profiles best match with the profile of *t*.

The remainder of this section is structure as follows. Section 4.4.1 introduces an approach for dataset interlinking recommendation. Section 4.4.2 presents the DRX architecture. Section 4.4.3 describes the DRX GUI and a case study. Section 4.4.4 presents the evaluation setup and the results of the experiments. Section 4.4.5 discusses and analyses the results. Finally, Section 4.4.6 presents the conclusions of this section.

4.4.1. An Approach to Dataset Interlinking Recommendation

The proposed approach has four main steps (see Figure **11**):

1. Data collection.
2. Dataset repository description.
3. Target dataset description.
4. Dataset interlinking recommendation.

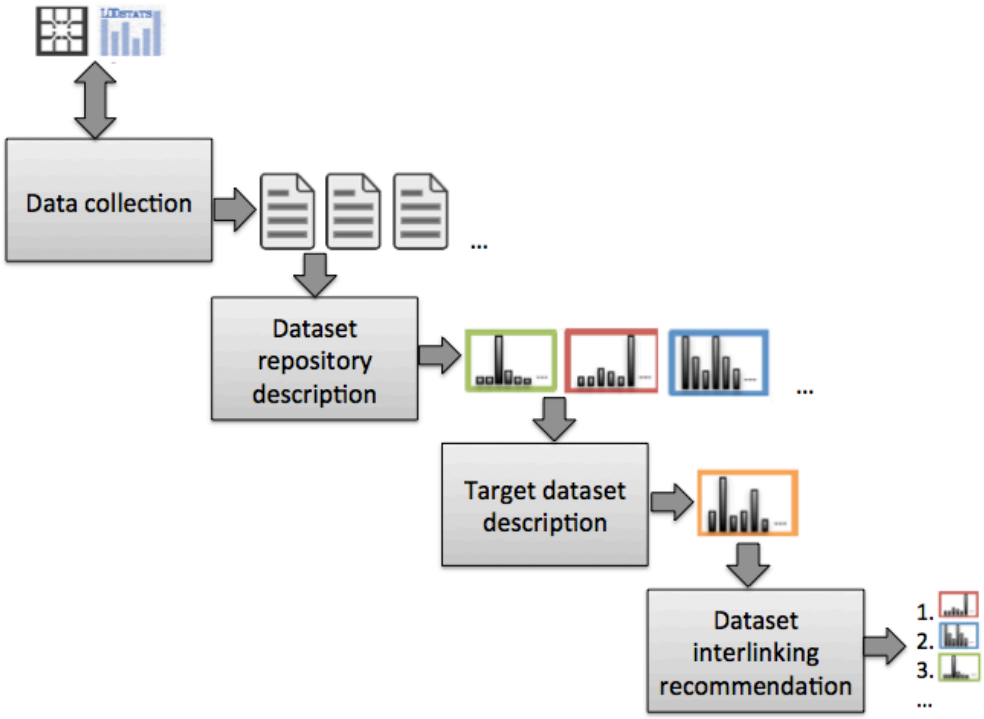


Figure 11: Dataset interlinking recommendation approach.

The first step aims in creating a document containing text literals or plain text for each LOD dataset. In order to reach this goal, it starts discovering metadata about the LOD datasets from LOD catalogs (such as the Mannheim[17] and DataHub[18]) as well as from manually submitted data. LOD catalogs typically stores metadata such as maintainer, author, SPARQL endpoint, relationships or linksets, VoID file, tags, license and resources. Then, metadata is inspected in order to obtain URIs that provides direct access to the data itself. Afterward, retrieved data that may be in many different formats passes for a process of filtering and sampling. Finally, this step creates a document containing the retrieved text; datasets with no text content are ignored, since the following step implements a technique that requires plain text.

The second step uses a profiling technique in order to create a fingerprints repository. Therefore, documents retrieved from a dataset are used to compute a description that characterizes the content stored in the dataset. The approach

[17] http://linkeddatacatalog.dws.informatik.uni-mannheim.de/

[18] https://datahub.io/

presented in this chapter implements the technique described in (KAWASE *et al.*, 2014), that generates dataset profiles or fingerprints from plain text.

The technique has five steps:

1. Extract entities from a given text literal.
2. Link the entities to English Wikipedia articles.
3. Extract the categories of the articles.
4. Follow the path from each extracted category to its top-level category and compute a vector with scores for the top-level categories thus obtained (such as agriculture, applied science, arts, belief, business, chronology, culture and so on).
5. Perform a linear aggregation in all dimensions of the vectors to generate the final profile, represented as a histogram for the 23 top-level categories[19] of the English Wikipedia.

The third step receives a source dataset *t* from a data publisher, and then applies the afore-explained profiling technique.

The last step, considers two strategies to provide recommendations for a given dataset *t*: *cluster-base*$_d$ and *profiling-based* strategies. The first strategy recommends datasets in the same cluster as *t* whereas the profiling based strategy considers all datasets identified by fingerprints. Independently of the strategy chosen, for a given dataset *t*, the dataset recommendation module outputs a list of datasets ordered by the probability of being interlinked. Assume that *t* is in cluster C_t. The cluster-based strategy creates a ranked list by taking into account only the distance between the fingerprint of *t* and the fingerprints of the other datasets in C_t. By contrast, the profiling-based strategy creates a recommendation list based on the distance between the fingerprint of *t* and the fingerprints of all other profiled datasets. We note that the distance function is a parameter of the recommendation algorithm; in the experiments, we adopted the cosine distance.

[19] https://en.wikipedia.org/wiki/Category:Main_topic_classifications

4.4.2.
DRX Architecture

The DRX tool is based on five modules, depicted in Figure **12**, which are distributed in three different layers: data acquisition, data processing and application. These modules perform five main tasks:

1. Collect data from datasets in the LOD cloud.
2. Process the data collected to create dataset profiles, called fingerprints.
3. Group fingerprints, using clustering algorithms.
4. Provide dataset recommendations.
5. Support browsing the dataset profiles.

The data acquisition layer includes the crawling module, which discovers metadata about the LOD datasets from LOD catalogs as well as from manually submitted data. The crawling module uses the CKAN API[20] to query metadata available in such catalogs.

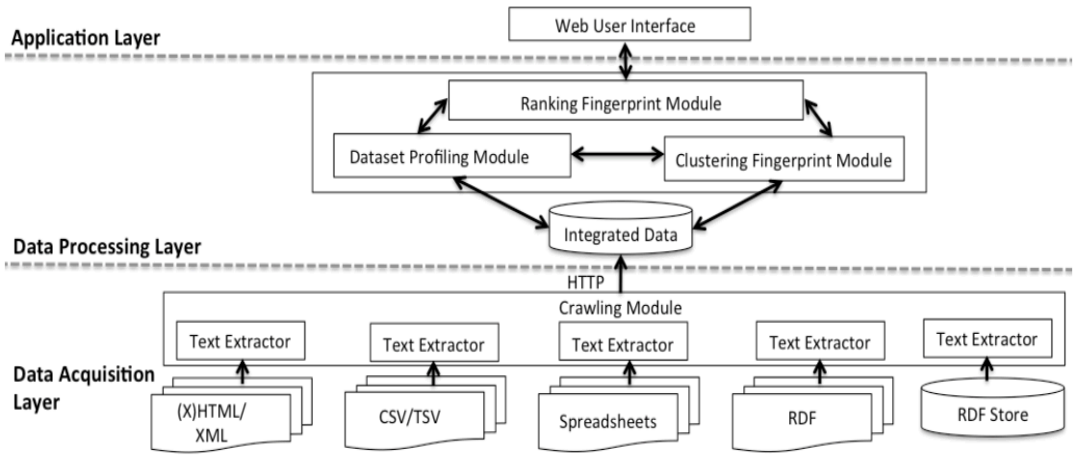


Figure 12: Architecture DRX tool.

[20] http://ckan.org/

Since data provided under each dataset can be in many different formats, the data acquisition layer also provides a set of specialized text extractors, that the crawling module uses to extract text literals or plain text from the datasets.

Once a dataset is located, the crawling module creates a document containing the retrieved text.

The data processing layer includes three main modules: profiling, clustering and ranking. The profiling module processes the documents retrieved from a dataset and computes a description that characterizes the content stored in the dataset. DRX implements the technique described in (KAWASE *et al.*, 2014), that generates dataset profiles or fingerprints from plain text.

The clustering module groups together fingerprints that are similar according to some distance measure. The goal is that fingerprints in the same cluster have a small distance from one another, while fingerprints in different clusters are at a larger distance one another. As is mentioned in (LESKOVEC, et al., 2014) Euclidean space's points are vectors of real numbers. Hence, the top-level categories of the English Wikipedia represent the number of dimension of the space. The DRX tool implements the X-Means clustering algorithm, which is part of the WEKA suite, and includes an efficient estimation of the number of clusters (HALL *et al.*, 2009; PELLEG et al., 2000).

The last module implements two strategies to provide recommendations for a given dataset *t*: *cluster-based* and *profiling-based* strategies.

4.4.3. DRX GUI and Case Study

The DRX[21] GUI allows the user to browse the LOD cloud by using dendrograms, tables and coordinate graphs and does not require any expertise in Semantic Web technologies or languages. The user may register a new source dataset *s* or select the source dataset *s* from those already in the LOD cloud. In either case, the user may then request recommendations for target datasets.

[21] http://drx.inf.puc-rio.br/

To illustrate how DRX works, we selected an independent dataset, rkb-explorer-newcastle[22], created jointly with other datasets under the ReSIST[23] project. These datasets are available through the RKBExplorer[24] Semantic Web browser that supports the Computer Science research domain. It combines information from multiple heterogeneous sources, such as published RDF sources, personal Web pages and databases in order to provide an integrated view of this multidimensional space.

In what follows, we refer to the five steps of our technique, introduced in Section 4.4.1. The goal of the first step is to collect text literals from data sources. If the user wants to consider a new dataset, he will first register it in the Mannheim catalog and submit its Mannheim URL entry to the tool. If the user selects an existing dataset, then the crawling module has already collected text literals, with the help of text extractors. In the case study, *the rkb-explorer-newcastle* dataset was crawled, using its SPARQL endpoint, to extract the text literal values of the *rdfs:label , skos:altLabel*, and *skos:prefLabel* properties.

The second step is carried out transparently to the user. Here, the text literals collected in the first step are used as input to the profiling module to create a description of the dataset content through a fingerprint. Table **8** presents the fingerprint generated for the case study dataset, where the 23-dimension vector shows peaks for “Society”, “Technology” and “Science”, categories that are strongly related to the data content that the *rkb-explorer-newcastle* dataset provides.

Table 8: Generated fingerprint for the rkb-explorer-newcastle dataset.

Category	**value**	**Category**	**value**
Agriculture	0	Humanities	0.16
Applied Science	0.02	Language	0.08

[22] http://linkeddatacatalog.dws.informatik.uni-mannheim.de/dataset/rkb-explorer-newcastle

[23] http://www.resist-noe.org/

[24] http://www.rkbexplorer.com/

Arts	0.02	Law	0.01
Belief	0.25	Life	0.09
Business	0.16	Mathematics	0.26
Chronology	0.20	Nature	0.25
Culture	0.13	People	0
Education	0.29	Politics	0.02
Environment	0.02	Science	0.49
Geography	0.01	Society	0.42
Health	0	Technology	0.42
History	0	-	-

To facilitate the exploration and selection of datasets, it is important to reduce the search space of datasets in the LOD cloud. Therefore, the third step generates clusters of datasets that share a certain similarity. The clustering module implements a simple interface that allows users to enter input parameters (such as the minimum and maximum number of clusters and the number of seeds and to execute the clustering process for all collected LOD datasets.

In the case study, we used a minimum of 8 clusters, since this is the number of categories of the LOD diagram[25]. The maximum number of clusters and the number of seeds were set to 10, (see Figure **13**).

Figure 13: DRX configuration Web Form.

[25] http://lod-cloud.net/

The result of the clustering process is represented as a dendrogram that allows users to navigate over the clusters and their respective members. For the case study, the clustering process generated 9 clusters; the *rkb-explorer-newcastle* dataset belongs to cluster #4, (see Figure **14**).

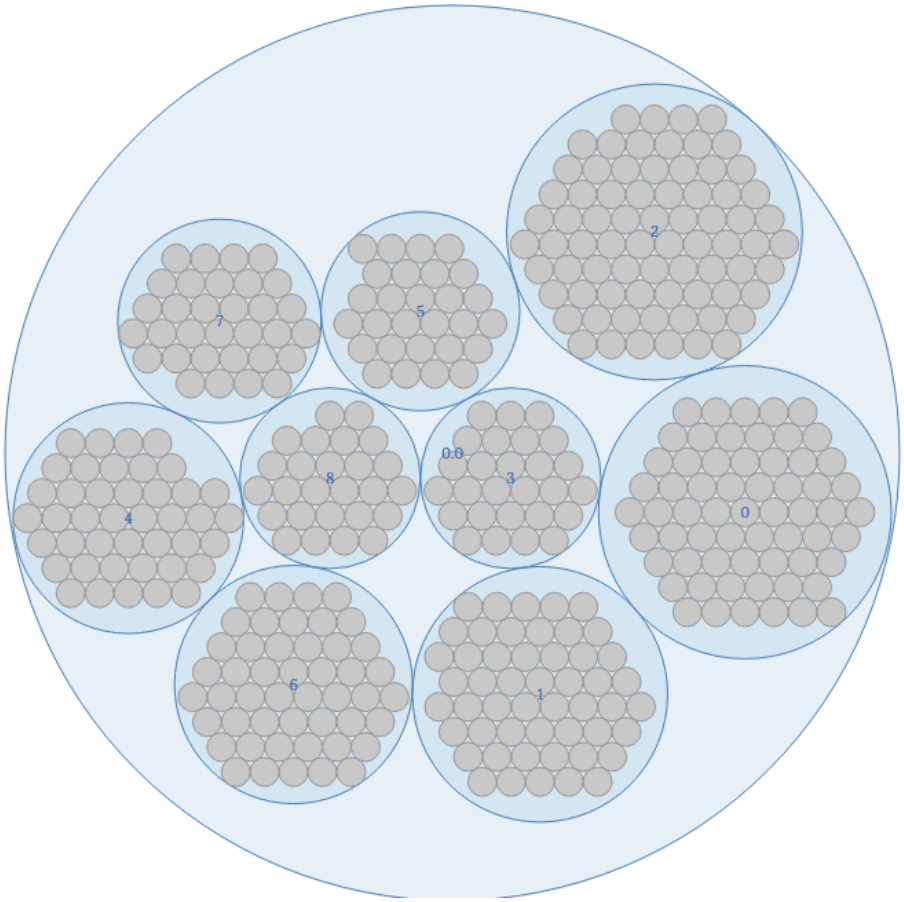
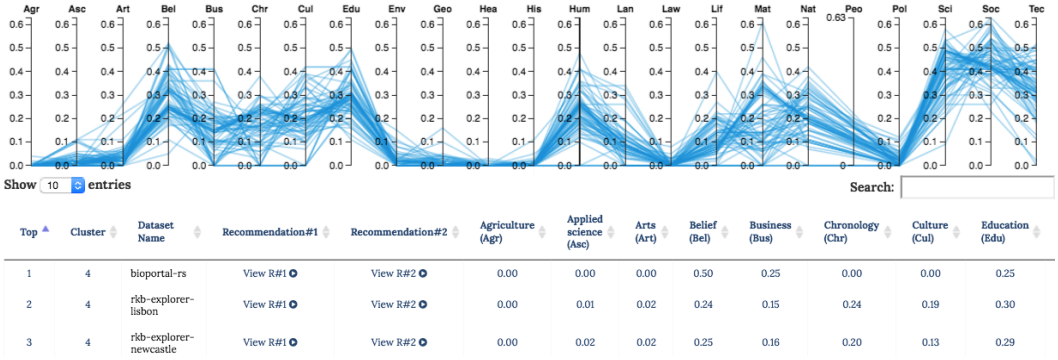


Figure 14: Result of the clustering depicted in a dendrogram.

The user interface also offers a zoom-in/out feature, which allows users to explore the members of each cluster in more detail; the user has to click inside a cluster area to zoom-in the cluster. For example, after zooming in cluster #4, we may observe some of its members, such as *rkb-explorer-roma , bioportalcheminf* and *rkb-explorer-newcastle*.

The user interface also provides a table with relevant information about the members of a selected cluster (see Figure **15**). This table offers column sorting and full text search. Each row shows the following information: the *"top"* column provides a dataset ranking based on the centrality of the datasets in the cluster; the *"cluster"* column represents the cluster membership; the *"dataset name"* column is a link to the dataset page in the Mannheim catalog; the *"Recommendation#1"*

column provides recommendations, for the dataset d_i on the selected row, from datasets of the same cluster as d_i; the *"Recommendation#2"* column provides recommendations, for the dataset d_i on the selected row, based on all datasets available and, finally, the other columns show the vector with the 23 top-level categories.



Top	Cluster	Dataset Name	Recommendation#1	Recommendation#2	Agriculture (Agr)	Applied science (Asc)	Arts (Art)	Belief (Bel)	Business (Bus)	Chronology (Chr)	Culture (Cul)	Education (Edu)
1	4	bioportal-rs	View R#1	View R#2	0.00	0.00	0.00	0.50	0.25	0.00	0.00	0.25
2	4	rkb-explorer-lisbon	View R#1	View R#2	0.00	0.01	0.02	0.24	0.15	0.24	0.19	0.30
3	4	rkb-explorer-newcastle	View R#1	View R#2	0.00	0.02	0.02	0.25	0.16	0.20	0.13	0.29

Figure 15: Detailed information of the members of a cluster.

For the case study, regarding cluster #4, Figure **15** shows detailed information of 10 members of cluster #4 out of a total of 67. The *rkb-explorer-newcastle* dataset was assigned the third position in the list, based on its centrality degree in the cluster.

Additionally, the user interface offers a feature to obtain interlinking recommendations. The user simply selects a dataset from the table in Figure **15**, then select one of the recommendation strategies and finally clicks on the corresponding cell of column. For the case study the first recommendation strategy was selected. Then, a table is displayed, containing a list sorted by ascending order of the cosine distance values (see Figure **16**). For the case study, 10 recommendations, of a total of 67 are displayed. Note that the top ten dataset recommendations are from the project that *rkb-explorer-newcastle* belongs to (see Figure **16**).

RECOMMENDATIONS FOR: RKB-EXPLORER-NEWCASTLE

List of recommendations for rkb-explorer-newcastle ordered by the cosine distance.
Rankings with distances closer to zero are more similar, while those with distances closer to 1 are more different.

Top	Cluster	Dataset Name	Distance	Gold standard
1	4	rkb-explorer-ieee	0.005	1
2	4	rkb-explorer-courseware	0.012	1
3	4	rkb-explorer-pisa	0.013	1
4	4	rkb-explorer-resex	0.014	1
5	4	rkb-explorer-webscience	0.017	0
6	4	rkb-explorer-curriculum	0.017	1
7	4	rkb-explorer-kaunas	0.017	0
8	4	rkb-explorer-budapest	0.018	1
9	4	rkb-explorer-eprints	0.019	1
10	4	rkb-explorer-epsrc	0.020	1

Showing 1 to 10 of 67 entries Previous 1 2 3 4 5 6 7 Next

Figure 16: List of dataset interlinking recommendations.

Finally, the user interface provides two tables containing the set of categories and entities extracted from the dataset in analysis. For the case study, Figure **17** and Figure **18** show the categories and entities obtained, respectively.

WIKIPEDIA CATEGORIES FOR: RKB-EXPLORER-NEWCASTLE

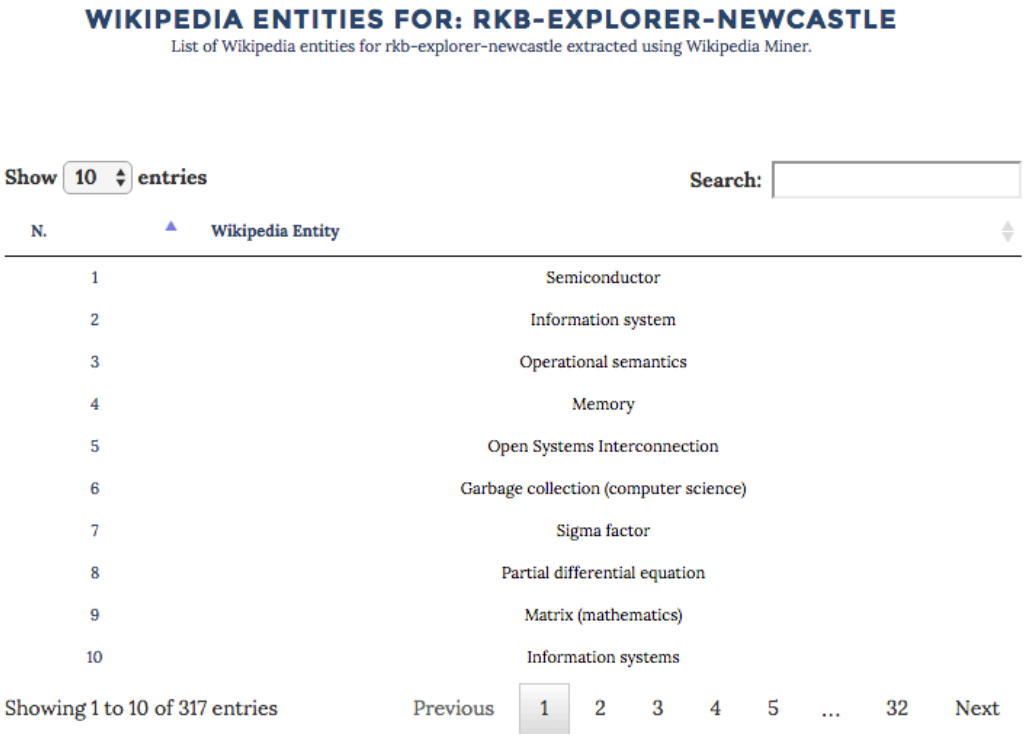
List of Wikipedia categories for rkb-explorer-newcastle.

Show 10 entries Search:

N.	Wikipedia Category
1	Functional_programming
2	Memory
3	Integrated_circuits
4	Servers
5	Computability_theory
6	Engineering_occupations
7	Design
8	Data_structures
9	Business_terms
10	Wireless_networking

Showing 1 to 10 of 636 entries Previous 1 2 3 4 5 … 64 Next

Figure 17: List of Wikipedia categories from rkb-explorer-newcastle dataset.



WIKIPEDIA ENTITIES FOR: RKB-EXPLORER-NEWCASTLE

List of Wikipedia entities for rkb-explorer-newcastle extracted using Wikipedia Miner.

Show 10 entries Search:

N.	Wikipedia Entity
1	Semiconductor
2	Information system
3	Operational semantics
4	Memory
5	Open Systems Interconnection
6	Garbage collection (computer science)
7	Sigma factor
8	Partial differential equation
9	Matrix (mathematics)
10	Information systems

Showing 1 to 10 of 317 entries Previous 1 2 3 4 5 ... 32 Next

Figure 18: List of Wikipedia entities from rkb-explorer-newcastle dataset.

4.4.4. Evaluation Setup

4.4.4.1. Data and Evaluation Metrics

The approach that DRX implements was assessed using data retrieved from the Mannheim catalog, a metadata repository for open datasets. Through the CKAN API, the catalog enables querying dataset metadata, including two multivalued properties (*relationships* and *extras*), which in turn allow data publishers to assert that a dataset links to another. Both properties were used to retrieve the linksets between datasets in the Mannheim catalog. During the crawling step, we retrieve all datasets that have at least one resource or data associated. In early 2016, the data collected amounts to 387 datasets that were profiled. However, during the evaluation, we considered a total of 165 datasets that were profiled and belong to the LOD diagram.

As in (CARABALLO *et al.*, 2013, 2014; EMALDI *et al.*, 2015; LEME *et al.*, 2013; LOPES *et al.*, 2013), linksets were used to define the gold standard for

the dataset interlinking recommendation approach of the DRX tool. That is, the evaluation consisted in removing the existing linksets between datasets and verifying to what extent DRX was able to include known interlinked datasets in the recommendation lists it produces. The performance of DRX was measured using the overall *Mean Average Precision* (MAP), defined in what follows.

Note that the gold standard comprises only the datasets listed in the Mannheim catalog for which the fingerprints could be computed. We deemed as unsuitable datasets with no associated data or with inaccessible endpoints, even if their metadata would indicate the existence of linksets. Clearly, there is no reason to recommend a dataset that is not accessible to participate in an interlinking process.

More precisely, let t be a source dataset for which one wants to recommend datasets to be interlinked with, and L_t be a ranked list of datasets recommended for t. Let G_t be the gold standard for t, i.e., the set of datasets that have linksets with t in the gold standard. A dataset d_i is relevant for t, in the context of G_t, iff there are linksets connecting d_i and t in G_t.

We then define:

– *Prec@k* (L_t), the precision at position k of L_t, is the number of relevant datasets in L_t until position k.

– *AveP* (L_t) is the average precision at position k of L_t, defined as:

$$AveP\ (L_t) = \sum_k Prec@k\ (L_t) / |\ G_t\ |.$$

Recall from Section 4.4.1, that the ranked list L_t of datasets recommended for t can be generated using two strategies: (i) *cluster-based*, that is, based on the datasets available within a cluster; and (ii) *profiling-based*, that is, based on all datasets available. The overall mean *average precision (MAP)* for these strategies is then defined in slightly different ways.

For the profiling-based strategy, we define:

– The *overall MAP* is the average of *AveP*(L_t) taken over the set of all datasets t

and, for the cluster-based strategy, we define:

- *MAP(C_i)*, the *Mean Average Precision* for C_i is the average of *AveP (L_t)* taken over the set of all datasets t in C_i
- The *overall MAP* is the average of *MAP(C_i)* taken over the set of all clusters C_i.

4.4.4.2. Results

We ran experiments considering the two recommendation strategies. For the cluster-based strategy, Figure **19** shows the overall MAP as a function of the number of clusters (in increments of 1). It indicates that the maximum value of overall MAP is 18.44%, when the number of clusters was equal to 11.

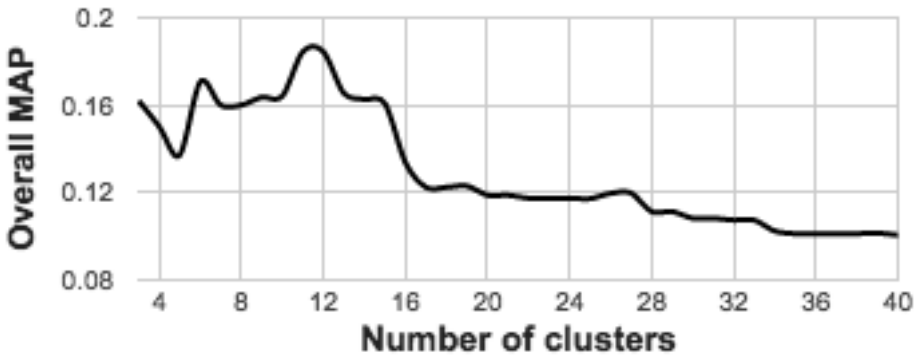


Figure 19: Strategy 1: Overall Mean Average Precision vs. number of clusters.

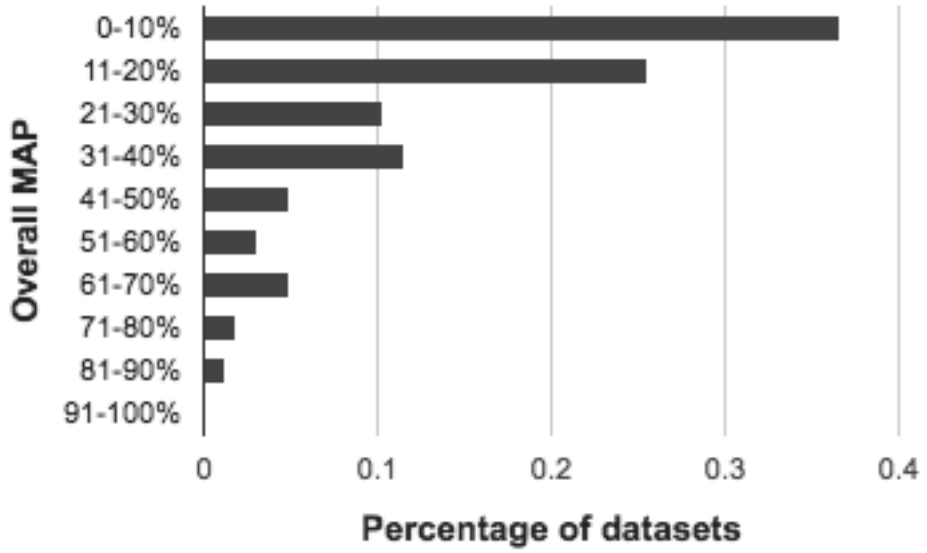


Figure 20: Strategy 2: Percentage of datasets vs. Overall Mean Average Precision.

Figure **19** indicates that the maximum MAP is obtained with 11 clusters, whereas the number of categories used to classify datasets in the LOD diagram is only 8. But if we construct just 8 clusters, our recommendation approach reaches an overall MAP of 16.0%, which is sub-optimal. That is, the LOD diagram is not a good starting point for our recommendation strategy. With only 8 clusters, many more non-relevant datasets end up being recommended, which decreases the overall MAP, as compared with the scenario that considers 11 clusters.

For the profiling-based strategy, Figure **20** presents the percentage of the total number of datasets for which the technique achieved a given MAP as a function of overall MAP intervals. It shows that this strategy: reached an overall MAP of 11-20% for 25% of the datasets; achieved an overall MAP higher than 20% for more than 37% of the datasets, reaching, in some cases, MAP values higher than 80%.

4.4.5. Discussion

The results reported in Section 4.4.4.2 should be assessed under several provisos, related to limitations of both the experiments and of the techniques the tool implements.

4.4.5.1. Profiling and Interlinking Issues

False positives. *Situation 1:* We first observe that we adopted as gold standard the LOD datasets represented in the Mannheim catalog that could be profiled. Furthermore, we considered that a source dataset d_i is correctly linked to a target dataset d_k iff the Mannheim entry for d_i contains a linkset for d_k. This may cause distortions on the results reported since DRX might correctly recommend a dataset d_m to be interlinked with d_i and no linkset is reported connecting d_i and d_m. This limitation has already been remarked in (EMALDI et al., 2015).

Therefore, d_m would be incorrectly considered as a false positive in the experiment, a situation that can only be uncovered by actually trying to interlink d_i and d_m, an expensive experiment that should be undertaken with care.

Situation 2: This scenario cannot be strictly considered as leading to a false positive, but the arguments are a continuation of the previous discussion. Consider again d_i and d_m, that is, DRX included d_m in the recommendation list for d_i. It might be the case that the user may try to interlink d_i and d_m without success because, although similar with respect to their profiles, these datasets contain different sets of resources that cannot be interlinked. The current implementation of the tool cannot automatically detect this situation, but its interface supports browsing the contents of a dataset so that the user may judge if the recommendation is likely to lead to interlinking d_i and d_m.

False negatives. *Situation 1:* The recommendation step depends heavily on the dataset profiling technique adopted. Let d_i be a dataset which is specific for a given area and, hence, whose profile has peaks for certain categories (such as agriculture, say). However specific d_i might be, it is common practice to interlink a dataset with a dataset d_m that contains generic data (such as the DBpedia). Since d_m is generic, its profile will tend to have high scores for most of the top categories.

Hence, DRX will probably not rank d_m high in the recommendation list for d_i (which we assumed is not a generic dataset). Therefore, d_m would be considered

a false negative in the experiment. This situation can be overcome by treating generic datasets (with high scores for most of the top categories) separately.

Situation 2: Conversely, d_i might be a dataset, which is more generic than d_m. Hence, DRX will probably not rank d_m high in the recommendation list for d_i. Therefore, d_m would be considered a false negative in the experiment.

4.4.5.2. Third party tools issues

As part of the dataset profiling step, we use the Wikipedia Miner (WM) to extract entities from a given text literal. WM achieves good precision (≈ 73%) and recall (≈75%) rates for entity recognition as reported in (ELLEFI et al., 2014). Also, in (FETAHU et al., 2014), they conducted a similar experiment where showed that misrecognized entities do not significantly impact in the resulting profile. In (FETAHU et al., 2014) also showed that the impact is related to the number of resources (sample size) extracted from a dataset to generate a proper profile. They considered that 10% of resources of a dataset produce a descriptive profile. For this reason, DRX considers only 10% of the resources from a dataset.

Another reason for the low impact of the misrecognized entities in the resulting profile is that the process does not solely consider the entities but their parent categories. The entity categories are grouped and only the categories over a given threshold are kept in the process. Thus, the misrecognized entities/categories will mostly probably not survive.

An example could be given by the following enriched text: "Pelé began playing for Santos at 15 and the Brazil national football team at 16. He won three FIFA World Cups." Santos can be a city or a Brazilian football team. In this example, we may expect that Santos can be recognized as a resource of the category Sports. However, suppose that Santos was misrecognized as a city resource. Whilst the others well recognized resources would contribute to the same categories (i.e. Sports), Santos, as a city, is expected to contribute to other categories not related. So, after running our process, the low contribution will not be considered, eliminating the contribution given by the misrecognized entity and hence its associated categories.

4.4.5.3. Examples of dataset profiles

A key factor to generate good dataset profiles is the availability of text literals. However, most datasets available in catalogs merely offer a small description with no associated data. So, a small sample of text literals may result in a not descriptive profile, increasing the probability of obtaining false positive recommendations.

Consider, for example, the *rkb-explorer-newcastle* dataset, which obtained a high MAP value (70%). The *rkb-explorer-newcastle* entry in the Mannheim Catalog is richly described by five different resource types: (1) XML Sitemap; (2) VoID File; (3) a resource Example; (4) an RDF file for download; and (5) an SPARQL endpoint that provides direct access to the entire content of the dataset. With such amount of information available, the creation of the dataset profile tends to be more accurate.

Inspecting the profile generated for the *rkb-explorer-newcastle* dataset, we verified that the recognized entities were strictly related to the information the dataset provides. A sample of the recognized entities is: Programmer, Computer Software, Functional Programming, and Neural Networks.

With enough resources available, DRX was able to generate a proper profile (fingerprint), selecting the best Wikipedia top-level categories to represent the dataset. In this case, DRX was able to generate a fingerprint with peaks at: Science, Technology and Mathematics, which are fully related to the dataset.

Unfortunately, as mentioned in (SCHMACHTENBERG *et al.*, 2014), only a few datasets have SPARQL endpoints available and rich descriptions. An example of a dataset with low MAP is the *Statusnet-doomicile-de*. As mentioned in (FETAHU *et al.*, 2014), and also in (SCHMACHTENBERG *et al.*, 2014), this dataset lacks information and resources. For example, the only file available for *Statusnet-doomicile-de* is an example resource. No SPARQL endpoint is available. Unfortunately, with so few resources available, it is impossible to properly cover the content of a dataset. Moreover, without an SPARQL endpoint,

DRX is unable to inspect its content. Hence, DRX does not generate a proper fingerprint resulting in a low MAP value.

Another issue found in datasets published in the Mannheim Catalog is that there are many datasets supposedly hacked. This is the case of: *HACKED BY SLAYERSHACKTEAM*, *admin*, *HacKeD By KingSkrupellos*, and many other datasets. The lack of quality, spam and curation lead us to low MAP values, not the method itself.

4.4.6. Conclusions

We proposed an approach implemented in a tool, called DRX, to assist data publishers in the process of dataset interlinking. DRX takes advantage of various methods including crawling, profiling, clustering and ranking modules to create ranked lists of datasets to be interlinked with a given dataset. The results obtained indicate that the proposed approach can indeed be used to facilitate the task of dataset interlinking in the LOD. They show that the profiling-based strategy achieves a better performance than the cluster-based strategy.

4.5. Tools comparison

This section presents an evaluation of the approaches introduced in this chapter.

4.5.1. Experiment Setup

4.5.1.1. Dataset, Ground Truth and Performance measures

The evaluation adopted a database with metadata of LD datasets. This database contains features such as: title, description, owner, vocabularies, properties, classes and linksets. In total, the database provides metadata about 99 LD datasets as well as 147 linksets. As a ground truth, we adopt the linksets. Basically, we classified dataset pairs into two classes: (i) (linked datasets) dataset

pairs that have a linkset between them; and (ii) (unlinked datasets) dataset pairs that are not connected by any linkset. In order to validate the tools in the classification task, we adopted the standard metrics introduces in Section 4.3.4.3 and Section 4.4.4.1.

4.5.1.2. Tools Setup

TRT requires the definition of a target context and uses link prediction measures to generate dataset interlinking recommendations. As the definition of the target context depends of the intuition of the user, it was randomly selected. Regarding the link prediction measures, we adopted the Preferential Attachment index, which obtained the best performance (CARABALLO *et al.*, 2013).

TRTML was evaluated using the J48 decision tree (Quinlan's C4.5 implementation), which achieved the best performance (CARABALLO *et al.*, 2014). DRX considered the profiling-based strategy, which achieved the best results (c.f. Section 4.4.4.2).

Table 9: Performance of the tools.

Measure	TRT	TRTML	DRX
Precision	11%	75%	26%
Recall	19%	11%	41%
F-measure	13%	19%	32%
MAP	22%	-	31%

4.5.2. Results

Table 9 presents the results for DRX, TRT and TRTML tools. The reader may observe that TRTML obtained the highest precision (75%). Table 9 also shows the coverage results. DRX achieved the highest recall (41%). Considering precision and recall under the F-measure, DRX obtained the highest (32%). Table 9 also shows the quality of the ranking, where DRX obtained the highest MAP (31%).

4.5.3. Analysis of the Features

In this section, we analyze how the tools support features that we regard as important for dataset interlinking recommendation. Specifically, we consider the following features: input data, degree of automation and dataset ranking (see Table **10**).

Table 10: Features of the tools.

Feature	TRT	TRTML	DRX
Input Data	Context & Link prediction Measure	VoID File	Data URL
Degree of Automation	Semiautomatic	Semiautomatic	Automatic
Dataset Ranking	Yes	No	Yes

Input data. By input data, we understand the data the user must provide to execute the tools. In the case of TRT, the user must provide a context and select a link prediction measure, based on his knowledge of the Linked Open Data (LOD) cloud. That is, to obtain proper dataset interlinking recommendations, the user must know the datasets in the LOD cloud and their underlying type of content to define a suitable context.

Regarding TRTML, the user must provide a VoID file describing the dataset: (1) the vocabularies used in the dataset, via the "void:vocabulary" VoID property; and (2) the classes and properties used in the dataset, via the "void:classPartition" and "void:propertyPartition" VoID properties. Therefore, the user must know the VoiD vocabulary and understand how the dataset was created.

As for DRX, the user must input only the URL of the dataset. Thus, DRX requires much less input data than the other tools.

Degree of Automation. A very important feature of any data extraction tool is its degree of automation. This is related to the amount of work left to the user during the process of generating dataset interlinking recommendations.

Regarding the degree of automation, TRT needs two inputs, a context and a link prediction measure, as already mentioned. Thus, the recommendation process is semi-automatic. TRTML provides a higher degree of automation. However, for this automation to be really effective, the user must provide a well-defined VoID file, as discussed above. Unfortunately, this is not true for a large fraction of the dataset available in the LOD. As stated in (SCHMACHTENBERG *et al.*, 2014), just 14.69% of the datasets published in the LOD have such descriptor.

DRX provides the highest degree of automation. The user must provide only the URL of the dataset he wants to obtain recommendation for. Then, the tool proceeds to sample the data, generate the fingerprint, and output the dataset interlinking recommendations.

Dataset Ranking. A very important feature of any dataset interlinking recommendation tool is to generate dataset rankings, which help reduce the search space that a user must face. Only TRT and DRX provide this feature. TRT outputs a ranked list, sorted by the probability of two dataset being linked, estimated using link prediction measures selected by the user. DRX also outputs a ranked list, sorted by the probability of two dataset being connected, estimated based on the cosine distance between the fingerprints.

4.6. Conclusions

In this chapter, we introduced three approaches to face the dataset interlinking problem, these approaches were implemented in three tools, named TRT, TRTML and DRX. We set up an experiment in order to validate the performance of each tool respect others. The experiment showed that TRTML obtained the highest precision (75%). In terms of recall, DRX performs better (41%) that the other tools. DRX also obtained the best f-measure (32%). The experiments also showed that in a direct comparison between DRX and TRT, DRX obtained a higher MAP value (31%) than TRT (22%). Moreover, when compared with TRT and TRTML with respect to their features, DRX presented a

better degree of automation and flexibility in terms of data input. Also, the features used by TRTML are not as discriminative (SCHMACHTENBERG *et al.*, 2014) as the features used by DRX, providing more information and leading to a better dataset recommendation.

5
Conclusions and Future Work

In this thesis, we focused on the development of approaches that address the dataset clustering and the dataset interlinking problems. In order to show the potential usefulness of the research, we implemented Web-based applications that follow the proposed approaches and tested them in real-world scenarios.

In Chapter 3, we addressed the problem of creating an automatic clustering of the datasets in the LOD cloud. Indeed, the manual classification of datasets in the LOD cloud, represented as a LOD diagram, has been well adopted in the Semantic Web Community and is the only available classification (SCHMACHTENBERG *et al.*, 2014; RODRIGUEZ, 2009a). However, new versions of the LOD diagram take several years to appear, which inhibits the consumption of recently published datasets. Therefore, we proposed an automatic clusterization procedure of the datasets in the LOD cloud based on dataset metadata (such as, description, title, and linksets). We implemented a tool that clusters the datasets in the LOD cloud and automatically labels each cluster, which effectively creates a way to automatically create LOD descriptions similar to the (manual) LOD diagram.

The results indicate that the best performing community detection algorithm is the GCE algorithm, with NMI and purity values of 0.57 and 0.42, respectively. Additionally, the mutual dependence between the communities generated using GCE and those from the ground truth is also not high, but, as discussed in Section 3.6, the lack of linksets between datasets in some domains, such as “Cross-Domain”, implies a need for the re-organization of datasets as well as the merging and splitting of communities. Additionally, a depth analysis of the manual labeling process showed that these labels considered as its classification criterion the nature of the data, whereas the automatic process relied on the contents of the datasets to generate the community labels. Finally, the experimental results showed that the proposed process automatically creates a clusterization of the

LOD datasets which is consistent with the traditional LOD diagram and that it generates meaningful descriptions of the dataset communities.

As for future work, we aim at: (a) implementing overlapping community detection algorithms; (b) identifying sub clusters exploring hierarchical clustering algorithms; and (c) applying the community description technique over sub clusters in order to generate more specialized descriptions.

In Chapter 4, we introduced three approaches for facing the dataset interlinking problem and presented the implementation of three tools that incorporate the ideas of the approaches.

For the dataset interlinking recommendation problem, we used in the first approach link prediction measures to estimate the likelihood of the existence of a link between datasets. A comprehensive evaluation of the introduced link prediction measures showed that TRT tool performs better, with respect to Recall, when the PA index is adopted. Moreover, in terms of MAP, the PA index should be adopted for smaller context sizes, while the RA index should be adopted for larger context sizes. The outcomes show that TRT can be used a dataset interlinking facilitator.

The second approach for dataset interlinking recommendation considered supervised learning algorithms that create a data model based on link prediction measures that explore a set of features (e.g. vocabularies, classes and properties) available for the datasets found in the metadata catalogs. An in-depth evaluation of the performance of the TRTML tool showed that J48 decision tree (Quinlan's C4.5 implementation) supervised learning algorithm achieved the best overall performance, when compared with the Multilayer Perceptron and the SVM algorithms. As a result TRTML showed the ability to reduce the effort of searching for related datasets in large data repositories.

The last approach used dataset profiling techniques and clustering algorithms in order to provide dataset interlinking recommendations. A complete evaluation using real-world dataset show that DRX has a potential to be used as a dataset interlinking facilitator.

As we proposed three different approaches to face the dataset interlinking recommendation problem, we setup an experiment to evaluate the performance of

each approach respect others. Results showed that DRX obtained the best f-measure (32%) and MAP (22%), respectively.

As for future work, we aim at: (a) supporting data publisher in selecting the target context when using the TRT tool; an alternative would be to use the dataset interlinking recommendations generated for both TRTML and DRX as a target context; (b) producing an automatic configuration of the link discovery frameworks based on the data and metadata available in the LD catalogs; and (c) running link discovery frameworks to validate the recommendations produced for our approaches.

Bibliography

BEEK, Wouter *et al.* LOD laundromat: A uniform way of publishing other people's dirty data. **International Semantic Web Conference – ISWC***, 2014*, Trento, Springer, v. 8796, p. 213–228, 2014.

BIZER, Christian. Evolving the Web into a Global Data Space. 2011, *Manchester ,Keynote at* **28th British National Conference on Databases - BNCOD**, p. 1, 2011.

BIZER, Christian; HEATH, Tom; BERNERS-LEE, Tim. Linked Data - The Story So Far. **Int. J. Semantic Web Inf. Syst.**, v. 5, n. 3, p. 1–22, mar. 2009.

CARABALLO, Alexander Arturo Mera; Nunes, Bernardo Pereira; Lopes, Giseli Rabello; Leme, Luiz André Portes Paes; Casanova, Marco Antonio. Automatic Creation and Analysis of a Linked Data Cloud Diagram. **Web Information Systems Engineering – WISE***, 2016*, Shanghai, Springer, v. 10041, 2016.

CARABALLO, Alexander Arturo Mera; Nunes, Bernardo Pereira; Lopes, Giseli Rabello; Leme, Luiz André Portes Paes; Casanova, Marco Antonio; Dietze, Stefan. TRT-A Tripleset Recommendation Tool. **International Semantic Web Conference – ISWC***,* 2013, Sydney, Springer, p. 105–108, 2013.

CARABALLO, Alexander Arturo Mera; Arruda, Narciso Moura; Nunes, Bernardo Pereira; Lopes, Giseli Rabello; Casanova, Marco Antonio. TRTML-A Tripleset Recommendation Tool Based on Supervised Learning Algorithms. **European Semantic Web Conference – ESWC 2014 Satellite Events**, Crete, Springer, p. 413–417, 2014.

ELLEFI, Mohamed Ben *et al.* Dataset recommendation for data linking: an intensional approach. **European Semantic Web Conference – ESWC 2016 Satellite Events***,* Crete, Springer, p. 36–51, 2016.

ELLEFI, Mohamed Ben *et al.* Towards Semantic Dataset Profiling. **Proceedings of the 1st International Workshop on Dataset PROFIling & fEderated Search for Linked Data co-located with the 11th Extended Semantic Web Conference**, Crete, 2014.

EMALDI, Mikel; CORCHO, Oscar; LÓPEZ-DE-IPINA, Diego. Detection of

Related Semantic Datasets Based on Frequent Subgraph Mining. **Intelligent Exploration of Semantic Data in Extended Semantic Web Conference**, 2015.

ERTÖZ, Levent; STEINBACH, Michael; KUMAR, Vipin. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. **Proceedings of the 2003 SIAM International Conference on Data Mining**. *Society for Industrial and Applied Mathematics*, San Francisco, p. 47–58, 2003.

FETAHU, Besnik *et al.* A Scalable Approach for Efficiently Generating Structured Dataset Topic Profiles. **Proceedings of the 1st International Workshop on Dataset PROFIling & fEderated Search for Linked Data co-located with the 11th Extended Semantic Web Conference**, Crete, p. 519–534, 2014

FORTUNATO, Santo. Community detection in graphs. **Physics Reports**, v. 486, n. 3-5, p. 75–174, fev. 2010.

GIRVAN, Michelle; NEWMAN, Mark E J. Community structure in social and biological networks. **Proceedings of the National Academy of Sciences**, v. 99, n. 12, p. 7821–7826, 2002.

GREGORY, Steve. Finding overlapping communities in networks by label propagation. **New Journal of Physics**, v. 12, n. 10, p. 103018, 2010.

HALL, Mark *et al.* The WEKA data mining software: an update. **ACM SIGKDD explorations newsletter**, v. 11, n. 1, p. 10–18, 2009.

JENTZSCH, Anja; CYGANIAK, Richard; BIZER, Chris. **State of the lod cloud**. Disponível em: <http://lod-cloud.net/state/>.

KAWASE, Ricardo *et al.* Exploiting the wisdom of the crowds for characterizing and connecting heterogeneous resources. **Proceedings of the 25th ACM conference on Hypertext and social media 2014**, Santiago, Springer, p. 1–4, 2014.

LALITHSENA, Sarasi *et al.* Automatic Domain Identification for Linked Open Data. **Web Intelligence (WI) and Intelligent Agent Technologies (IAT)**, 2013, IEEE/WIC/ACM International Joint Conferences, v. 1. p. 205–212, 2013.

LEE, Conrad *et al.* Detecting highly overlapping community structure by greedy clique expansion. **arXiv preprint** *arXiv:1002.1827*, 2010.

LEME, Luiz André P Paes *et al.* Identifying Candidate Datasets for Data Interlinking. **International Conference on Web Engineering**, 2013, Berlin, Springer, p. 354–366, Berlin.

LESKOVEC, Jure; RAJARAMAN, Anand; ULLMAN, Jeffrey David. **Mining of massive datasets**, Cambridge University Press, 2014.

LIU, Haichi *et al.* Identifying Linked Data Datasets for sameAs Interlinking Using Recommendation Techniques. **International Conference on Web-Age Information Management**, 2016, Springer, p. 298–309, 2016.

LOPES, Giseli Rabello; LEME, Luiz André P Paes; *et al.* Recommending Tripleset Interlinking through a Social Network Approach. **Web Information Systems Engineering – WISE**, 2013, Nanjing, Springer, p. 149–161, 2013.

LÜ, Linyuan; JIN, Ci-Hang; ZHOU, Tao. Similarity index based on local paths for link prediction of complex networks. **Physical Review** *E*, v. 80, n. 4, p. 046122, out. 2009.

MANNING, Christopher D; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. **Introduction to information retrieval**, Cambridge university press Cambridge, 2008. v. 1.

NGOMO, A.C.N.; AUER, Sörer. Limes-a time-efficient approach for large-scale link discovery on the web of data. **Proceedings of IJCAI**, p. 2312–2317, 2011.

NIKOLOV, Andriy; D'AQUIN, Mathieu. Identifying relevant sources for data linking using a Semantic Web index. **WWW2011 Workshop: Linked Data on the Web (LDOW 2011)***,* Hyderabad, 2011.

NUNES, Bernardo Pereira *et al.* Complex matching of rdf datatype properties. **International Conference on Database and Expert Systems Application**, 2013, Springer, p. 195–208, 2013.

PELLEG, Dan; MOORE, Andrew W. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. **Proceedings of the Seventeenth International Conference on Machine Learning***,* Stanford, 2000, Pat Langley, v.1 p. 727-734 2000.

RODRIGUEZ, Marko A. A graph analysis of the Linked Data cloud. **arXiv preprint** *arXiv:0903.0194*, 2009a.

SCHMACHTENBERG, Max; BIZER, Christian; PAULHEIM, Heiko. Adoption of the Linked Data Best Practices in Different Topical Domains. **International Semantic Web Conference – ISWC**, Riva del Garda, 2014, Springer, p. 245–260, 2014.

TUMMARELLO, Giovanni *et al.* Sig. ma: Live views on the Web of Data. **Web Semantics: Science, Services and Agents on the World Wide Web**, v. 8, n. 4, p. 355–364, 2010.

VOLZ, Julius *et al.* Silk-A Link Discovery Framework for the Web of Data. **LDOW**, v. 538, 2009.

XIE, Jierui; KELLEY, Stephen; SZYMANSKI, Boleslaw K. Overlapping community detection in networks: The state-of-the-art and comparative study. **ACM Computing Surveys (CSUR)**, v. 45, n. 4, p. 43, 2013.