Enterprise Information Integration

On Discovering Links using Genetic Programming

Andrea Cimmino

University of Sevilla, Spain

Doctoral Dissertation
Supervised by Dr. Rafael Corchuelo

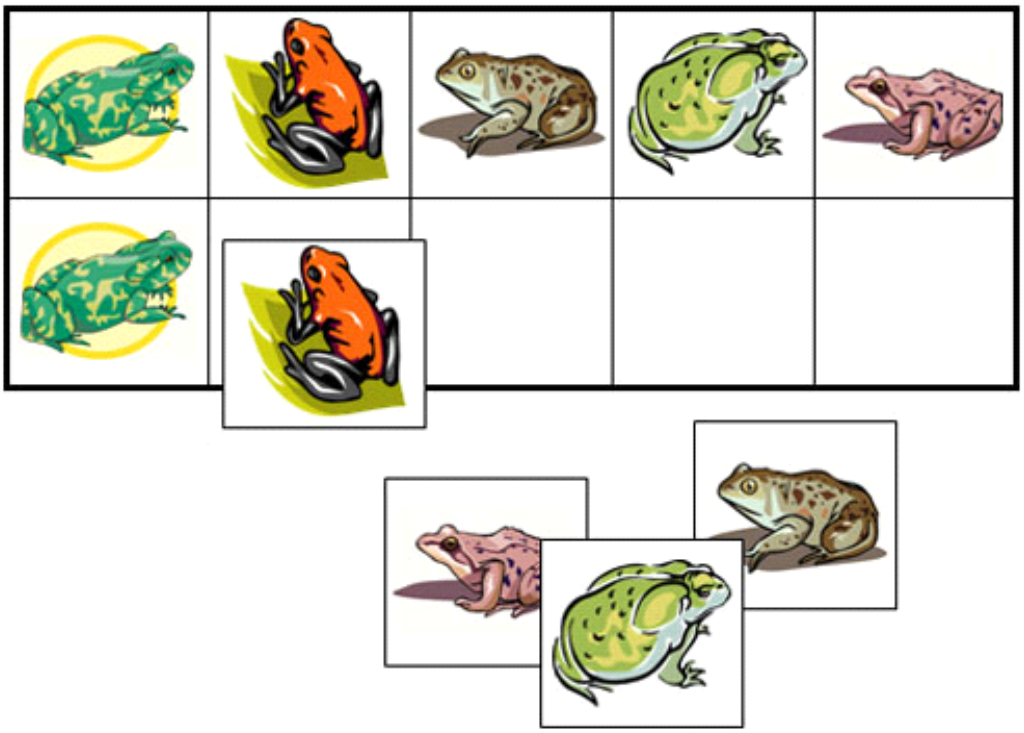
September, 2019

First published in July 2019 by
The Distributed Group
ETSI Informática
Avda. de la Reina Mercedes, s/n
Sevilla, E-41012. SPAIN

http://www.tdg-seville.info
contact@tdg-seville.info

Classification (ACM 1998): D.2.12 [Software Engineering] Inter-operability: Data mapping; H.3.1 [Information Storage and Retrieval] Content Analysis and Indexing: Linking data; H.3.5 [Information Storage and Retrieval] On-line Information Services: Data sharing.

Support: Supported the Spanish R&D&I programme by means of grants TIN2013-40848-R and TIN2016-75394-R.



Frog linking game, all ages allowed.

Dedicated to those who held a light during my long nights

Contents

List of figures

List of tables

List of algorithms

Acknowledgements

Whilst these lines flow out of my fingers I realise how I am settling the end of an era, which may have seem so distant as faint lights but yet is here. It is with sadness that I finish my PhD, because it has been one of the greatest vital experiences I have ever had. I am well aware that I would not be here without the support, love, and comprehension of those who stood by me during this treacherous, yet marvellous, path.

I would like to thank my parents Mercedes and Andrea for their entire life of support and love, as well as, the rest of my family. I also want to warmly show my gratitude to Isabel for understanding all these weekends of working. I cannot forget to mention Mouly, Lucia, Carlos, Patricia, and many others who fed me with their positivism and backed me up during these years.

During my PhD I had two main "research" families. On the one hand, many thanks to the TDG family who raised me during this years: to my colleagues with whom I had such a fruitful discussions, to the post-docs who showed us that the treacherous path had an end as well, and to Dr. Ruiz and Dr. Rivero who sharpened my mind. On the other hand, many thanks to my new family, the OEG, who welcomed me from the first day and made me feel at home, beloved, and supported. Special thanks go to Dr. García-Castro for his support and teachings, which I hope there still will be plenty.

Last, but not least, I wish to thank my supervisor, Dr. Rafael Corchuelo. I will never have enough words to express my gratitude to him. To me he is one of the best mentors someone may have. He inspired me from an early age and encourage me to always chase my ideas trusting my instinct and without being afraid of failing, and also to never forget to enjoy the haunting. Rafael Corchuelo has always bet for me, help me out, and taught me how to be a researcher and a better person.

Abstract

Both established and emergent business rely heavily on data, chiefly those that wish to become game changers. The current biggest source of data is the Web, where there is a large amount of sparse data. The Web of Data aims at providing a unified view of these islands of data. To realise this vision, it is required that the resources in different data sources that refer to the same real-world entities must be linked, which is they key factor for such a unified view. Link discovery is a trending task that aims at finding link rules that specify whether these links must be established or not. Currently there are many proposals in the literature to produce these links, especially based on meta-heuristics. Unfortunately, creating proposals based on meta-heuristics is not a trivial task, which has led to a lack of comparison between some well-established proposals. On the other hand, it has been proved that these link rules fall short in cases in which resources that refer to different real-world entities are very similar or vice versa.

In this dissertation, we introduce several proposals to address the previous lacks in the literature. On the one hand we, introduce Eva4LD, which is a generic framework to build genetic programming proposals for link discovery; which are a kind of meta-heuristics proposals. Furthermore, our framework allows to implement many proposals in the literature and compare their results fairly. On the other hand, we introduce Teide, which applies effectively the link rules increasing significantly their precision without dropping their recall significantly. Unfortunately, Teide does not learn link rules, and applying all the provided link rules is computationally expensive. Due to this reason we introduce Sorbas, which learns what we call contextual link rules.

Resumen

Las empresas que desean establecer un precedente en el panorama actual tienden a recurrir al uso de datos para mejorar sus modelos de negocio. La mayor fuente de datos disponible es la Web, donde una gran cantidad es accesible aunque se encuentre fragmentada en islas de datos. La Web de los Datos tiene como objetivo dar una visión unificada de dichas islas, aunque el almacenamiento de los mismos siga siendo distribuido. Para ofrecer esta visión es necesario enlazar los recursos presentes en las islas de datos que hacen referencia a las mismas entidades del mundo real. Link discovery es el nombre atribuido a esta tarea, la cual se basa en generar reglas de enlazado que permiten establecer bajo qué circunstancias dos recursos deben ser enlazados. Se pueden encontrar diferentes propuestas en la literatura de link discovery, especialmente basadas en meta-heurísticas. Por desgracia comparar propuestas basadas en meta-heurísticas no es trivial. Por otro lado, se ha probado que estas reglas de enlazado no funcionan bien cuando los recursos que hacen referencia a dos entidades distintas del mundo real son muy parecidos, o por el contrario, cuando dos recursos muy distintos hacen referencia a la misma entidad.

En esta tesis presentamos varias propuestas. Por un lado, Eva4LD es un framework genérico para desarrollar propuestas de link discovery basadas en programación genética, que es un tipo de meta-heurística. Gracias a nuestro framework, hemos podido implementar distintas propuestas de la literatura y comprar justamente sus resultados. Por otro lado, en la tesis presentamos Teide, una propuesta que recibiendo varias reglas de enlazado las aplica de tal modo que mejora significativamente la precisión de las mismas sin reducir significativamente su cobertura. Por desgracia, Teide es computacionalmente costoso debido a que no aprende reglas. Debido a este motivo, presentamos Sorbas que aprende un tipo de reglas de enlazado que denominamos reglas de enlazado con contexto.

Chapter 1

Introduction

This chapter introduces our PhD work. It is organised as follows: Section §1.1 introduces the context of our research work; Section §1.2 presents an overview of the related work; Section §1.3 presents the hypothesis that has motivated our work and states our thesis; Section §1.4 summarises our main contributions; Section §1.5 sketches the collaborations that we have conducted throughout the development of this dissertation; finally, Section §1.6 describes the structure of this document.

1.1 Research context

The feasibility of many emerging business relies on the availability and inter-operability of suitable on-line datasets [3]. The Web of Data provides business with islands of data (availability) that can be transparently used as required by the business models (inter-operability). The Web of Data builds on the Linked-Data principles that support the idea that resources within different datasets that refer to the same real-world entities must be linked so as to facilitate data inter-operability [10]. Link rules are intended to help linking resources automatically relying on different kind of relations, e.g., spatial or temporal coverage, we focus only on *owl:sameAs* that relates resources representing the same real-world entity.

The whole process to link two datasets is known as link discovery; which consists in two phases: the former aims at generating link rules that encode whether two resources should be linked, and the latter, focuses on efficiently applying these rules. Learning link rules consist in selecting transformation and similarity metrics that are applied to the literals of the data properties of two resources to check if they can be considered similar enough; if they are, then the input resources are linked; otherwise, they are kept apart. Furthermore, linking two datasets entails performing a Cartesian product of their resources, and then, applying a link rule over all the resource pairs to check whether they should be linked. The product is usually very large, due to this reason is why researchers have proposed so-called blocking techniques; which aim at reducing the number of resource pairs that have to be checked when applying a rule.

Learning link rules is a task that requires to explore a number of potential solutions that is very large; due to this reason researchers have approached the problem relying of meta-heuristics proposals. This kind of proposals are inspired by nature and their main feature is their capability to explore in parallel a large space of solutions, and generally, manage to find a suitable solution. Therefore, it is not unusual to find proposals that learn link rules following the genetic programming approach, which belong to a particular kind of meta-heuristics that are known as evolutive proposals.

The literature provides several genetic programming based proposals for link discovery [21, 28, 29, 41, 42, 44, 49, 51]. These proposals generate link rules relying on several heuristics, and then, refine such rules using a function that is able to quantify the effectiveness of the link rules. Genetic

programming builds on a fixed template that encodes its main functionality, which is always the same, and a set of heuristics that change depending on the specific implementation, which are intend to refine the rules.

Despite the large number of link discovery proposals, as far as we know, these proposals have not been compared under the same experimental circumstances. Some surveys presented a comparison of these proposals from a conceptual point of view, and some other just gather the results from the original papers of the proposals; furthermore, each proposal was executed following different method of evaluation, e.g., k-fold or exhaustive, and with different number of examples. Furthermore, the authors did not rely on statistical analysis to support claims regarding the comparison of the proposals. In particular, genetic programming proposals lack a comparison survey due to the lack of a generic framework to build specific implementations, and then, compare these implementations; thus, comparing them under the same experimental circumstances is unfeasible. Therefore, it is not clear which genetic programming proposal works better in the scenarios of the literature.

We have also realised that the link rules generated by current proposals usually fall short when linking resources that refer to different real-world entities that have a similar representation, or when link rules have to deal with resources that have very different representations but which refer to the same real-world entities. Our experience proves that been able to address these two challenges is paramount to link real-world scenarios. The reason for this drawback is the fact that link rules compare the data properties of the resources been linked, but consider no related resources in such linking process preventing them to capture restrictions about the contextual information of such resources.

1.2 Related work

Link discovery is a task equivalent to record linkage in the relational databases field, due to this reason in the following sub-sections we first present an overview of the literature concerning link discovery surveys in relational databases. Then, we describe surveys of link discovery proposals in the Web of Data. Next, we present research works related to methodologies defined for Ontology Matching, that is different from link discovery. Finally, since we focus on genetic programming proposals we aim at presenting how the most well-known proposals compared their results with other proposals.

1.2.1 Link discovery in relational databases

In the literature of relational databases, the task of record linkage addresses the same problem of link discovery [16, 30]. The most well-known surveys are the ones proposed by Köpcke and Rahm [33], Köpcke and others [34], and Elmagarmid and others [24].

Köpcke and Rahm [33] presented a survey analysing a large number of proposals, namely: BN [37], MOMA [54], SERF [8], Active Atlas [53], MARLIN [9], Multiple Classifiers System [58], Operator Trees [12], TAILOR [23], FEBRL [15], STEM [32], Context Based Framework [14]. Köpcke and Rahm grouped them according to their features: a) Entity type, whether proposals are able to deal with data modelled as trees (XML) or only tabular data; b) Blocking methods, used by the proposals to deal with large amount of data; c) Matching algorithms, that specify whether two records refer to the same real-world entity; d) Whether proposals are able to combine different matching algorithms; and e) Whether the approach is manual, supervised, semi-supervised, or unsupervised. Then, the authors compare the evaluations results published in the corresponding research papers of each proposal in terms of F_1. Finally, the authors conclude that: a) Current proposals have high effectiveness, efficiency, generality and low manual effort; b) There is a trend in using supervised and hybrid approaches; c) Frameworks evaluations use diverse methodologies, measures, and datasets that hinder the comparison of their results and, therefore, they see a strong need for standardised benchmarks, make available the prototype implementations, and data used by the proposals to learn.

Köpcke and Rahm proposed a framework to build and evaluate machine learning proposals applied to entity resolution [34]. Using that framework the same authors compared the effectiveness of several classifiers [36], namely: Decision tree, Logistic regression, SVM, Multiple learning, Baseline strategy. In their paper, the authors first analyse the datasets used in the evaluation of several entity resolution papers introducing those that are going to be used by them. Then authors analyse proposals that follow a manual approach, as a result they compare the effectiveness obtained in terms of Precision, Recall, and F_1. Next, the authors do the same for supervised proposals in the same datasets, as a result they compare their effectiveness in terms of labelling effort and F_1. Finally, the authors concluded that: a) SVM learning proposals are suitable especially in the bibliographic datasets from the benchmark; b) the E-commerce datasets are especially challenging since they

require large training data sets. c) The best scalability was achieved by the proposals implemented with the PPJoin+ framework, however scalability is something to be still improved.

Elmagarmid and others [24] presented a survey that leans towards entity matching approaches rather than specific proposals. In their work the authors characterize in a taxonomy the different approaches that proposals may follow, classifying a large number of them. Unfortunately, neither experimental data is presented not the results obtained in the specific datasets by the classified proposals.

1.2.2 Link discovery in the Web of Data

In the link discovery literature there are two surveys, by Nentwig and others [39] and Soru and Ngomo [50]. In addition, each year the OAEI initiative organises a contest in which one challenge is link discovery. Every year, a summary report is published; in this section, we focus on the 2017 report.

Nentwig and others [39] presented a paper analysing a wide number of proposals and their features, namely: RiMOM [52], KnoFuss [45], AgreementMaker [20], Silk [56], CODI [27], LIMES [40], LogMap [31], SERIMI [4], Zhishi.links [47], SLINT+ [43]. Nentwig and others first compared their different features: a) Data formats supported by the proposals, i.e., SPARQL, RDF, OWL; b) Whether they are manual, supervised, or unsupervised; c) Runtime optimization, if proposals use blocking or filtering approaches; d) Matching strategies, the approach followed by the proposals; e) Whether proposals support post-processing tasks, e.g., clerical review; f) Whether proposals support parallel processing; g) If proposals have GUI; and h) source code or prototype availability. Then, the authors report on the datasets in which the proposals were evaluated. Finally, the authors compare the results in terms of the F_1 score. Their conclusions are: a) most proposals focus on simple property-based matching proposals instead of using the ontological context with structural matchers; b) Efficiency must be improved by means of parallel execution, filtering proposals and blocking; c) To assess the comparison of the effectiveness and efficiency of the different proposals it would be valuable to have a common set of benchmarks to test the proposals, even with the same hardware if possible.

Soru and Ngomo [50] compared the effectiveness of several machine learning approaches, namely: Multi-Layer Perceptron, Logistic Regression, Linear SMO, Decision Table, J48, Linear Regression, Random Trees, Linear

SVM, Polynomial-3 SVM, and Naïve Bayes. Soru and Ngomo aimed at answering three main questions: a) Which of the machine learning approaches achieves the average best F_1 score; b) Which of the machine learning approaches is most robust against noise; c) Which of the approaches is the most time-efficient. To find the answers, the authors executed several proposals and compared their results in terms of average F_1. As a result, the authors state the following answers for such questions: a) Random Trees and Multi-Layer Perceptron obtained the best results; b) Noisy datasets suggest that Multilayer Perceptron are the most robust to be used; c) All approaches scale well but random trees were the fastest in a few datasets. d) As a general conclusion, the authors suggest using Multi-Layer Perceptron as they achieve acceptable runtimes are robust against noise.

1.2.3 Ontology matching methodologies

Ontology matching is the task of aligning the terms defined in an ontology with the terms of another, although this task may look similar to link discovery they are quite different tasks. In ontology matching the number of elements to be aligned is much less that the number of resources to be linked in link discovery. In addition, one may know beforehand the terms related to a type, but different resources of the same type may have different data properties or literals. Therefore, both tasks have differences in their goals and how to achieve them.

Nevertheless they are related since one approach followed by ontology matching proposals is to perform a link discovery task between the data instances of two types, each one of a different ontology. Building on the number of links generated, the proposal can decide if both types should be aligned. This approach is known as data interlinking, that is the name of the OAEI challenge from which most of the link discovery proposal use their datasets; due to the fact that data interlinking and link discovery are essentially the same.

The yearly OAEI initiative contest has a challenge of data interlinking, in which all participant proposals are analysed and compared [1]. Some participants are Silk [56], LogMap [31], SERIMI [4], or Zhishi.links [47]. Participating proposal are evaluated in terms of runtime, number of instances linked, Precision, Recall, and F_1. Each year new datasets to participate in this challenge are released, although the ones released in 2010 are the most used in the literature. In the OAEI results the features of the proposals are not compared, they only focus on effectiveness. In addition, the yearly reports always

use a Bernoulli significance test to support the claims regarding which proposal is better. As a result of each yearly paper, a ranking is computed and they report the top-three proposals. The OAEI does not count with a proper methodology, but some guidelines that were devised by García-Castro and Gómez-Pérez [26] are followed to carry out the experiments.

1.2.4 Genetic programming based proposals

The generation of link rules has being extensively addressed in the literature by proposal based on genetic programming algorithms [21, 28, 29, 41, 42, 44, 49, 51]. We classify these proposals into three groups, namely: a) Supervised proposals that require annotations to be provided with; b) Active Learning proposals that implement an algorithm that requests annotations from the user as needed; c) Unsupervised proposals that do not require any annotations; and d) Hybrid proposals that include some external logic in addition to the basic genetic programming algorithm. In the genetic programming algorithms, the link rules are represented as trees of functions; therefore, they typically use tree-manipulation functions.

Supervised proposal. Isele and Bizer [28] presented Genlink, their proposal follows a 2-fold validation applied to a narrowed version of the datasets. In their experiments they compared the results obtained in terms of iterations, training runtime and F_1, and validation F_1 with the proposal by de Carvalho and others [21]. However, the results used in their comparison were taken from the original paper by de Carvalho and others instead of computing both proposals under the same experimental conditions. The datasets used in by Isele and Bizer [28] and de Carvalho and others [21] were the ones released by the OAEI contest in 2011, and the ones from the benchmark by Chaudhuri and others [13]. In addition, Genlink is the only genetic programming proposal that introduces an algorithm to pair the data properties of the resources from different datasets; which consist in applying a Levenshtein function to all the possible label pairs and keeping only those with a perfect score, i.e., 1.00. Genlink is part of the well-known link SILK suite [56].

de Carvalho and others [21] evaluated their proposal following a 10-fold validation on a narrowed version of the dataset, however instead of relying on $k-1$ training sub-sets, the authors used one for training and another for validation, thus, a kind of 2-fold validation rather than 10-fold. Authors reported their results in terms of runtime, training F_1, and validation F_1. Some of the datasets used were extracted from the benchmark by Chaudhuri and others [13], and others generated with the synthetic data generator of FEBRL [15]. Finally, no comparison with other proposals was reported.

Besides their proposal, de Carvalho and others [22] also presented an article related to genetic programming algorithms that analysed the impact of the different parameter turnings.

Active learning proposal. Ngomo and Lyko [41] presented Eagle. The authors performed a first experiment in which they applied a handcrafted link rule and used its results as a gold standard, then relied on Eagle to obtain a rule as close as possible. Then, authors performed some experiments on the datasets used by FEBRL [15] and MARLIN [9] and they compared the results in terms of the F_1 score and the running time. The datasets in which a link rule was used as gold standard are Dailymed-Drugbank and DBpedia-LinkedMDB, the other experiments were run in the dataset of DBLP-ACM extracted from the benchmark devised by Köpcke and others [35]. Eagle is part of the well-known LIMES suite [40].

Isele and others [29] presented an upgrade of Genlink called ActiveGenlink. ActiveGenlink works in the same way of its previous version but includes an active learning algorithm based on a query strategy that relies on a heuristic to iteratively select meaningful annotated examples from available data and required users to accept or decline such examples in order to keep learning link rules.

Freitas and others [25] presented an upgrade of the proposal by de Carvalho and others [21] by including an algorithm of active learning. The active learning is based on a committee of functions that provides the regular proposal of de Carvalho and others with annotated examples, in case that the committee is not able to reach a verdict a user must make the call of whether use or not the annotated examples.

Unsupervised proposal. Borges and others [11] proposed an upgrade for the proposal by de Carvalho and others [21] to make it unsupervised; unfortunately, this upgrade is intended to work solely on bibliography datasets.

Nikolov and others [44] presented an unsupervised proposal. The novelty of their proposal is its custom objective function that takes the size of the link rules into account by combining the results of a Pseudo-F_1 function with a function called neighbourhood growth, which was devised by the authors. The authors evaluated their proposal, first, using narrowed versions of the OAEI benchmark of 2010 and 2011 from the data interlinking track, and then, with some curated datasets. In addition, the authors relied on some preprocessing techniques to clean the data (such as removing stop words), and

to index data so the linking process was faster. The proposal by Nikolov and others [46] is available as part of the well-known framework KnoFuss.

Hybrid proposal. Singh and Sharan [49] proposed an adaptive genetic programming algorithm. Their proposal relies on a heuristic that can be applied to any genetic programming algorithm that learns effective link rules in less time than regular genetic programming algorithms. The heuristic is based on several policies to prune the link rules learnt that are unappealing by two means: the size of the link rules and the F_1. The authors evaluated their proposal relying on the 2010 OAEI datasets and the CORA dataset by Chaudhuri and others [13].

Sun and others [51] proposed an entity resolution approach combined with genetic programming. Their proposal uses a genetic programming algorithm that relies on custom functions to select, cross, mute, replace, and create link rules. The link rules that this algorithm handles have a fixed structure defined by the authors, therefore, preventing them to grow in size. The proposal relies on the F_1 score as the fitness function. The authors evaluated their proposal using a narrowed version of the CORA dataset [13], and compared their results with the proposals by Isele and Bizer [28] and de Carvalho and others [21]. Nevertheless, the authors did not execute the literature proposals, but just compared the results reported by Isele and Bizer [28] and [21].

1.2.5 Discussion

Research proposals like the ones by Elmagarmid and others [24], Köpcke and Rahm [33], Köpcke and others [34], Nentwig and others [39], and Soru and Ngomo [50] are suitable for practitioners and researchers since they analyse the features of the proposals and this is useful to select proposals based on a set of requirements. However, selecting proposals based on their effectiveness is something that requires an analysis that relies on statistical significance tests to support its claims. The OAEI results may provide an overview of the effectiveness of different ontology matching proposals to practitioners; unfortunately, this task is slightly different from link discovery and may fall short when linking two datasets.

Regarding the genetic programming proposals from the literature [21, 28, 29, 41, 42, 44, 49, 51], none of them actually compares its results with other proposal proposals in the same experimental conditions. The authors usually compare their results with the ones reported in the articles of the other proposals, and furthermore, no ranking of the results obtained is computed.

1.3 Research rationale

According to Yuhanna and others [57], data integration is a paramount tasks for both emergent and consolidated companies. Vargas and others [55] also claim that the current problem is moving from a web of data islands to a real Web of Data. Link discovery is a key task to realise this vision. In a truly Web of Data, one may have access to a piece of data and navigate through the related data, independently from whether they are hosted by the same provider or a different provider. Belissent and others [6] also pointed out how having this Web of Data is clearly profitable for the business models of many companies. Furthermore, Belissent and others [7] have identified ten key-factors in the technology related to processing data: the fifth one state that the Web should be a large graph of data stored in distributed environments, but correctly linked so that it can be viewed as a unique dataset. This argumentation leads to the following hypothesis:

> *The trends that companies are adopting nowadays suggest that exploiting the data in the Web is becoming paramount for their business. Nevertheless, current data are fragmented in isolated datasets that must be linked, effectively and efficiently, so as to relate the data regarding the same real-world entities that are available in disparate islands of data. Companies who wish to play a major role in this world are interested in proposals to link data.*

Assuming this hypothesis, our analysis of the related work reveals two key-points. The former one is that link discovery proposals based on meta-heuristics, such as the genetic programming, have not being properly compared due to the lack of a framework that enables a fair experimental environment; and thus, it is not clear which proposal behaves the better. The latter is that the existing link discovery proposals lack a proper approach to improve the precision of the integration results since they do not exploit contextual data. These two points clearly justifies working on a framework that helps implement genetic programming proposals, and allows a fair comparison between them, and enhancing these proposals by significantly improving the their precision without degrading their recall. This argumentation leads to the following thesis, which we prove in this dissertation:

> *On the one hand, it is possible to fairly compare link discovery proposals based on meta-heuristics, and rank them building on their*

results to conclude which behaves better depending on the scenario. On the other hand, it is possible to significantly improve the precision of the link discovery proposals without a significant drop in their recall. We conjecture that combining link rules and exploiting contextual information is the key.

1.4 Summary of contributions

Our main contributions are the following:

Eva4LD: it is a generic genetic programming framework to implement specific proposals that perform link discovery. It is easily configurable by end-users and brings the opportunity to test new implementations of genetic programming proposals for link discovery, and, on the other hand, allows to compare the different proposals under the same experimental conditions. Bringing a fair environment to compare them. We have one journal paper presenting Eva4LD that is currently under review.

Teide: it is an approach that combines several link rules to exploit the context of each resource. It aims at improving the precision without dropping recall. Teide aims at addressing resources that are different but have very similar representations, or on the contrary, have different representations but refer to the same real-world entities. We have one workshop [19], and two conference papers [17, 18].

Sorbas: this proposal aims at learning contextual link rules, instead of only applying rules efficiently like Teide. We have proved that this approach is as effective as Teide, but more computationally efficient. We have one journal paper about Sorbas that is currently under review.

1.5 Collaborations

The motivation to work on this dissertation comes from two national research projects. The former is ISIDORO, in which we worked on data integration and we studied the link discovery proposals. We realised the lack of a common environment to fairly compare the proposals, and then, we found out that link rules had this drawback when facing similar resources that are different, and different resources that are the same. Later, in the project VORTEX, we worked on combining the link rules to improve their precision

without dropping their recall; and Teide was devised for this task. Then, we realised that Teide was computationally expensive, so we devised Sorbas which aims at fixing this drawback. In addition, during the PhD we collaborated with Dr. Carlos Rivero from the Rochester Institute of Technology in New York, USA. Our combined effort settled the pillar ideas on top of which Teide and Sorbas were built.

1.6 Structure of this dissertation

This dissertation is organised as follows:

- The introduction comprises this chapter, in which we motivate our research work and conclude that there are two main needs to cover. The former is to have a fair experimental environment to compare genetic programming proposals, and the latter, that link rules fall short when addressing some resources to be linked.
- Chapter §2 reports on our generic framework to implement specific genetic programming proposals for link discovery, and thus, a fair experimental environment. In addition, it provides a statistical ranking of the main proposals from the literature and three additional proposals that we have devised.
- Chapter §3 describes our proposal called Teide, which combines link rules as so to their precision is significantly improved without a significant drop in their recall.
- Chapter §4 presents our proposal Sorbas, which learns contextual link rules to link two datasets. In terms of effectiveness the link rules learnt by Sorbas are the same of how Teide combines and applies the link rules. In terms of efficiency, however, once the rules are learnt by Sorbas this approach outperforms Teide.
- Appendix §A reports on the computing facility we used in our experiments and the different scenarios in which we relied to perform our experimentation.
- Appendix §B introduces our running examples in which we explain the different proposals and showcase some examples.

Chapter 2

Eva4LD: A Genetic Framework

This chapter introduces our framework called Eva4LD, which helps practitioners to implement genetic programming-based link discovery proposals. It is organised as follows: Section §2.1 introduces the context of our framework; Section §2.2 presents some preliminary concepts; Section §2.3 describes and explains our framework; Section §2.4 describes several instantiations of our framework; Section §2.5 explains and describes the experiments conducted with our implementations; and finally, Section §2.6 recaps on the conclusions drawn from our experiments.

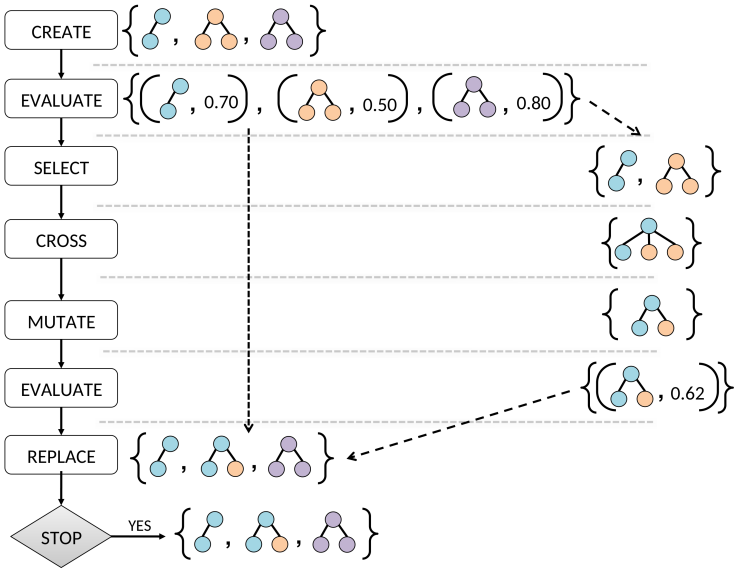


Figure 2.1: *A sample genetic programming workflow.*

2.1 Introduction

We have devised a framework called Eva4LD that is intended to help implementing link discovery proposals based on supervised genetic programming algorithms from the literature. These algorithms aim at generating an initial set of potential solutions for a problem, and then at refining such solutions relying on a function that quantifies how good one solution is. The different steps in which solutions go through in these algorithms are meant to refine elements that conform the solutions, so the overall quality of the solutions improves.

Genetic programming algorithms are inspired by genetic algorithms, the main difference between these two are the solutions they handle. In the former, solutions are meant to be executable models; in the latter the solutions are flat data structures that represent a solution for a maximization/minimization problem. In the context of link discovery, genetic programming algorithms handle link rules represented like trees of functions as potential solutions. The function used to evaluate the quality of such link rules usually involves evaluating the link rules against a set of reference links.

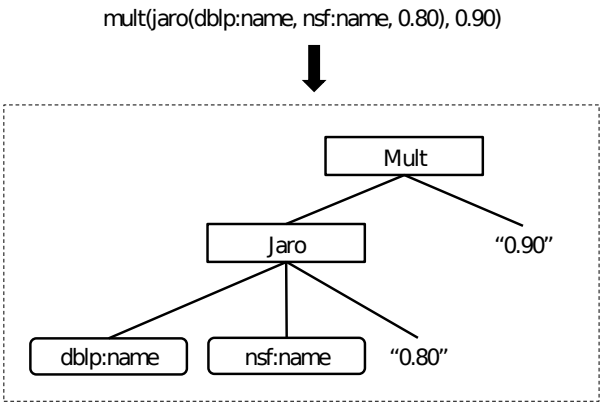


Figure 2.2: *Expressing a link rule as a tree solution.*

Figure §2.1 shows a sample workflow: it first creates a set of link rules, then, it evaluates how good these rules are by means of a score in range $0.00 \ldots 1.00$. Next, some of these link rules are selected and combined by means of crossover and mutation functions. As a result new link rules are obtained, which are evaluated as well. Finally, the new link rules replace some of the initial ones; if a stop criterion has being reached, the algorithm stops returning the set of link rules computed in so far; otherwise this process is repeated until a stop criterion is met.

Our framework is composed of a template, which is a generic algorithm that provides a harness to implement a variety of genetic programming algorithms by instantiating a number of variation points. By implementing the different variation points different proposals may be built.

2.2 Preliminaries

In this section, we introduce the concepts required to explain our framework.

Definition 2.1 (Dataset) *A dataset is a set of RDF triplets that follows the W3C specification [5]. The triplets consist of a subject and a predicate that are IRIs, and an object that can be either a literal or an IRI. The IRI of the subjects uniquely identify resources within the dataset, the IRI of the predicates refers to a namespace of a vocabulary (e.g. schema.org, owl, or foaf). Predicates are typically called data properties when they associate subjects with*

objects that are literals; they are typically called object properties when they associate subjects with objects that are IRIs of other subjects.

Example 2.1 *Our running example from Section §B.1 presents two sample datasets that are based on the DBLP and the NSF datasets, which are described in Appendix §A.2. The resources are depicted in greyed boxes whose shapes encode their classes (i.e., the value of property* rdf:type*), the properties are represented as labelled arrows, and the literals are encoded as strings.*

Definition 2.2 (Link) *A link is a triplet that relates the subject IRIs of two resources. The predicate that links the IRIs can be* owl:sameAs *if both resources refer to the same real-world entities, or on the contrary,* owl:differentFrom *if they refer to different entities.*

Example 2.2 *Assuming the datasets depicted in Section §B.1, the following are sample links:*

(dblp:*weiwang*, owl:sameAs, nsf:$weiwang_1$),
(dblp:*weiwang*, owl:differentFrom, nsf:$weiwang_2$),
(dblp:*weiwang*, owl:differentFrom, nsf:binliu),
(dblp:binliu, owl:sameAs, nsf:binliu),
(dblp:binliu, owl:differentFrom, nsf:$weiwang_1$)),
(dblp:binliu, owl:differentFrom, nsf:$weiwang_2$)),
(dblp:euzenat, owl:differentFrom, nsf:$weiwang_1$),
(dblp:euzenat, owl:differentFrom, nsf:$weiwang_2$), *and*
(dblp:euzenat, owl:differentFrom, nsf:binliu).

Definition 2.3 (Scenario) *A scenario is a triplet that consist of two datasets containing resources that refer to the same real-world entities, and a set of links that relate the resources within these datasets.*

Example 2.3 *We define the scenario Researchers relying on our running example described in Section §B.1 based on DBLP and NSF datasets,* D_1 *and* D_2 *respectively, and the set of links* L

(dblp:*weiwang*, owl:sameAs, nsf:$weiwang_1$),
(dblp:*weiwang*, owl:differentFrom, nsf:$weiwang_2$),
(dblp:*weiwang*, owl:differentFrom, nsf:binliu),
(dblp:binliu, owl:sameAs, nsf:binliu),
(dblp:binliu, owl:differentFrom, nsf:$weiwang_1$),
(dblp:binliu, owl:differentFrom, nsf:$weiwang_2$),

$(\mathtt{dblp{:}euzenat}, \mathtt{owl{:}differentFrom}, \mathtt{nsf{:}weiwang}_1)$,
$(\mathtt{dblp{:}euzenat}, \mathtt{owl{:}differentFrom}, \mathtt{nsf{:}weiwang}_2)$, *and*
$(\mathtt{dblp{:}euzenat}, \mathtt{owl{:}differentFrom}, \mathtt{nsf{:}binliu})$.

The Researchers scenario is defined by the tuple (D_1, D_2, L).

Definition 2.4 (Link Rule) *A link rule is a model that given two resources determines whether they refer to the same real-world entities. Link rules build on aggregate metrics, string metrics, and transformation metrics that are used to compare the values of a subset of data properties. When a rule is applied between two resources, the data properties within the rule are replaced by their actual values in the context of these resources. After such replacement, the link rule is evaluated obtaining a value between* 0.00*, entailing that resources are not linked, or* 1.00*, entailing that such resources are linked by a triplet (*IRI_1*,* owl:sameAs*,* IRI_2*), where* IRI_1 *is the subject IRI of the first resource and* IRI_2 *of the second.*

Example 2.4 *Consider the datasets DBLP and NSF depicted in Section §B.1, and the link rule:*

$$
\begin{aligned}
\mathtt{r}\text{: } \mathtt{link}(A, R) \text{ if } & \mathtt{rdf{:}type}(A) = \mathtt{dblp{:}Author},\\
& \mathtt{rdf{:}type}(R) = \mathtt{nsf{:}Researcher},\\
& N_A = \mathtt{dblp{:}name}(A), N_R = \mathtt{nsf{:}name}(R),\\
& S_1 = \mathtt{jaro}(N_A, \mathtt{normalize}(N_R)),\\
& \frac{S_1 - 0.80}{1.00 - 0.80} > 0.
\end{aligned}
$$

To link the resources within these datasets, first we need to consider the pairs of dblp:Author *and* nsf:Researcher *resources, namely:*

$(\mathtt{dblp{:}weiwang}, \mathtt{nsf{:}weiwang}_1)$,
$(\mathtt{dblp{:}weiwang}, \mathtt{nsf{:}weiwang}_2)$,
$(\mathtt{dblp{:}weiwang}, \mathtt{nsf{:}binwliu})$,
$(\mathtt{dblp{:}binliu}, \mathtt{nsf{:}weiwang}_1)$,
$(\mathtt{dblp{:}binliu}, \mathtt{nsf{:}weiwang}_2)$,
$(\mathtt{dblp{:}binliu}, \mathtt{nsf{:}binwliu})$,
$(\mathtt{dblp{:}euzenat}, \mathtt{nsf{:}weiwang}_1)$,
$(\mathtt{dblp{:}euzenat}, \mathtt{nsf{:}weiwang}_2)$, *and*
$(\mathtt{dblp{:}euzenat}, \mathtt{nsf{:}binwliu})$.

Then for each of these pair of resources the link rule r *is instantiated by replacing the data properties with the literals that these resources have. For instance, the pair* $(\mathtt{dblp{:}weiwang}, \mathtt{nsf{:}weiwang}_2)$ *instantiates the link rule as*

$\mathtt{jaro}(\text{“}WeiWang\text{”}, \mathtt{normalize}(\text{“}Wang, Wei\text{”}))$. *The result such expression is* 1.00*, therefore, evaluating* $\mathtt{r}$ *obtained a score of* 1.00 *as well. As a result,* $\mathtt{r}$ *would return tuple* $(\mathtt{dblp}{:}weiwang, \mathtt{owl}{:}\mathtt{sameAs}, \mathtt{nsf}{:}weiwang_2)$ *for* $\mathtt{dblp}{:}weiwang$ *and* $\mathtt{nsf}{:}weiwang_2$.

Definition 2.5 (Chromosome) *A chromosome is a tuple that relates a link rule and its effectiveness score.*

Example 2.5 *Figure §2.2 shows a link rule can be mapped or expressed as a tree. A chromosome using such rule is defined as* $(\mathtt{r}, 0.47)$*, where* 0.47 *is the effectiveness score of* $\mathtt{r}$.

2.3 Template

In this section, we present the template of our framework, which is sketched in Algorithm §2.1. It provides a harness in which the variation points are written in capital letters. Later, we provide additional details on how to instantiate the variation points.

The template gets as input a scenario (D_1, D_2, L) and, as a result, it provides a set of link rules R learnt. In addition, the template is fed with some arguments: a maximum population size s, the crossover probability p_c, and the mutation probability p_m.

The different steps performed in our template shown in Algorithm §2.1 are: i) Create a set of chromosomes C known as parents; ii) Create a set of chromosomes called offspring by selecting link rules from the chromosomes within the parents set, selected link rules are combined by means of crossover and mutation variation points to build new link rules, which are then transformed into new chromosomes to fill the offspring set; iii) Create a new parents set by combining the offspring set and the previous parents set. iv) Repeat this steps until a stop criterion is reached.

Example 2.6 *Let us work on the Researchers scenario from our previous examples, which consist of the datasets DBLP and NSF described in Section §B.1, and the following set of links* L*:*

$(\mathtt{dblp}{:}weiwang, \mathtt{owl}{:}\mathtt{sameAs}, \mathtt{nsf}{:}weiwang_1)$,
$(\mathtt{dblp}{:}weiwang, \mathtt{owl}{:}\mathtt{differentFrom}, \mathtt{nsf}{:}weiwang_2)$,
$(\mathtt{dblp}{:}weiwang, \mathtt{owl}{:}\mathtt{differentFrom}, \mathtt{nsf}{:}binliu)$,
$(\mathtt{dblp}{:}binliu, \mathtt{owl}{:}\mathtt{sameAs}, \mathtt{nsf}{:}binliu)$,
$(\mathtt{dblp}{:}binliu, \mathtt{owl}{:}\mathtt{differentFrom}, \mathtt{nsf}{:}weiwang_1)$,

```
GenericTemplate(S) : R
  parameters s, p_c, p_m

  – Step i)
  C: = ∅
  R :=CREATE(s)
  for r in R do
     M :=ComputeConfusionMatrix(D_1, D_2, L, r)
     c :=(r, EVALUATE(M, r))
     C :=C ∪ {c}
  end

  – Step ii)
  R :=∅
  while ¬STOP(C) do
     C′ := ∅
     while |C′| < s do
        R :=SELECT(C)
        if random(0.00, 1.00)> p_c then
           R :=CROSSOVER(R)
        end
        if random(0.00, 1.00)> p_m then
          R :=MUTATE(R)
        end
        for r in R do
            M :=ComputeConfusionMatrix(D_1, D_2, L, r)
           c′ :=(r, EVALUATE(M, r))
           C′ :=C′ ∪ {c′}
        end
     end
     – Step iii)
     C :=REPLACE(C, C′)
  end
  R :=rules(C)
end
```

Algorithm 2.1: *Template for genetic-programming algorithms.*

$(\text{dblp:binliu}, \text{owl:differentFrom}, \text{nsf:}weiwang_2)$,
$(\text{dblp:euzenat}, \text{owl:differentFrom}, \text{nsf:}weiwang_1)$,
$(\text{dblp:euzenat}, \text{owl:differentFrom}, \text{nsf:}weiwang_2)$, *and*
$(\text{dblp:euzenat}, \text{owl:differentFrom}, \text{nsf:binliu})$.

We provide as arguments for the maximum population size s *a value of* 3, *and* 0.40 *for the crossover* p_c *and mutation* p_m *probabilities, respectively.*

First, due to the value of s *a number of three link rules are generated by the variation point CREATE, for instance:*

r_1: $\text{link}(A, R)$ if $\text{rdf:type}(A) = \text{dblp:Author}$,
$\text{rdf:type}(R) = \text{nsf:Researcher}$,
$N_A = \text{dblp:name}(A), N_R = \text{nsf:name}(R)$,
$S_1 = \text{jaro}(N_A, N_R)$,
$\frac{S_1 - 0.92}{1.00 - 0.92} > 0.$

r_2: $\text{link}(A, R)$ if $\text{rdf:type}(A) = \text{dblp:Author}$,
$\text{rdf:type}(R) = \text{nsf:Researcher}$,
$A_A = \text{dblp:affiliation}(A), N_R = \text{nsf:name}(R)$,
$S_1 = \text{levenshtein}(A_A, N_R)$,
$\frac{S_1 - 0.21}{1.00 - 0.21} > 0.$

r_3: $\text{link}(A, R)$ if $\text{rdf:type}(A) = \text{dblp:Author}$,
$\text{rdf:type}(R) = \text{nsf:Researcher}$,
$N_A = \text{dblp:name}(A), N_R = \text{nsf:name}(R)$,
$A_A = \text{dblp:affiliation}(A), U_R = \text{nsf:university}(R)$,
$S_1 = \text{cosine}(N_A, N_R), S_2 = \text{jaccard}(A_A, U_R)$,
$\text{average}(\frac{S_1 - 0.37}{1.00 - 0.37}, \frac{S_2 - 0.87}{1.00 - 0.87}) > 0.$

Then, such rules are evaluated obtaining an effectiveness score of 0.63, 0.27 *and* 0.79, *respectively, by means of the variation point EVALUATE. Using these link rules and their scores three new chromosomes are computed,* c_1 *that is* $(r_1, 0.63)$, c_2 *that is* $(r_2, 0.27)$, *and,* c_3 *that is* $(r_3, 0.79)$. *All chromosomes conform the set of parents* $C = \{(r_1, 0.63), (r_2, 0.27), (r_3, 0.79)\}$.

The variation point EVALUATE relies on a confusion matrix M *to evaluate the effectiveness of a rule. The confusion matrix* M *is computed by means of* ComputeConfusionMatrix *which links using* r *the resources of* D_1 *and* D_2. *By comparing the links obtained by* r, *i.e.,* L', *and the ones in* L

this method computes the true positives as $tp = |L' \cap L|$*; false positives as* $fp = |L' \setminus L|$*; true negatives as* $tn = |\{l \mid owl{:}differentFrom \in l \wedge l \in L\}| - fp$*; and the false negatives as* $fn = |\{l \mid owl{:}sameAs \in l \wedge l \in L\}| - tp$*. As a result, the confusion matrix is defined as the tuple* $M = (tp, fp, tn, fn)$*. Relying on this matrix EVALUATE computes an effectiveness score.*

Second, the offspring set C' *is computed. The SELECT variation point picks some rules from the parents set* C*, for instance,* r_1, r_2 *and* r_3*. Then all rules are combined to build new ones by means of variation points* CROSSOVER *and* MUTATE *with a probability of* 0.40 *for both variation points. In both cases, a random variable uniformly distributed in interval* $[0.00, 1.00]$ *is sampled; if the results is greater than the crossover or the mutation probabilities, then these variation points are executed. As a result, a set of rules* R *is computed, containing:*

r_4: $link(A, R)$ if $rdf{:}type(A) = dblp{:}Author$,
$rdf{:}type(R) = nsf{:}Researcher$,
$A_A = dblp{:}affiliation(A), N_R = nsf{:}name(R)$,
$S_1 = cosine(A_A, normalize(N_R))$,
$\frac{S_1 - 0.21}{1.00 - 0.21} > 0.$

r_5: $link(A, R)$ if $rdf{:}type(A) = dblp{:}Author$,
$rdf{:}type(R) = nsf{:}Researcher$,
$N_A = dblp{:}name(A), N_R = nsf{:}name(R)$,
$A_A = dblp{:}affiliation(A), U_R = nsf{:}university(R)$,
$S_1 = levenshtein(N_A, N_R), S_2 = jaccard(A_A, U_R)$,
$average(\frac{S_1 - 0.37}{1.00 - 0.37}, \frac{S_2 - 0.87}{1.00 - 0.87}) > 0.$

r_6: $link(A, R)$ if $rdf{:}type(A) = dblp{:}Author$,
$rdf{:}type(R) = nsf{:}Researcher$,
$N_A = dblp{:}name(A), N_R = nsf{:}name(R)$,
$S_1 = jaro(N_A, N_R)$,
$\frac{S_1 - 0.92}{1.00 - 0.92} > 0.$

Notice that some are new link rules like r_4 *and* r_5*, and others are not, like* r_6*. Finally, these rules are transformed into chromosomes by evaluating them by means of EVALUATE variation point and included in the offspring set, in this case,* $C' = \{(r_4, 1.00), (r_5, 0.94), (r_6, 0.63)\}$.

Third, a new set of parents is computed by combining the offspring set C' *and the parents set* C *by means of the REPLACE variation point. For instance,*

if $C = \{(r_1, 0.63), (r_2, 0.27), (r_3, 0.79)\}$ *and* $C' = \{(r_4, 1.00), (r_5, 0.94), (r_6, 0.63)\}$, *the new parent set is* $C = \{(r_4, 1.00), (r_5, 0.94), (r_2, .027)\}$.

Finally, each time a set of parents is computed, the template checks whether a stop criterion is met and stops accordingly. In the case of our example, a solution obtained a score of 1.00 *in its effectiveness score, which is a widely extended stop criterion in genetic programming algorithms. As a result, in our example the algorithm would output* $R = \{r_4, r_5, r_2\}$. *Further sections detail the variation points of our template.*

2.3.1 Variation point: CREATE

The variation point CREATE aims at generating link rules, which are built following different heuristics depending on the implementation of this variation point. We define $\mathsf{CREATE}(s) = R$ where s is the maximum population size, and R is set of link rules. This function generates a number of s link rules.

Example 2.7 *In our running example this variation point generates the following link rules, e.g.,* $\{r_1, r_2, r_3\}$, *from our previous examples. The rules are created selecting random string metrics and combining them with data properties and thresholds selected randomly as well. In addition, some rules may combine two string metrics with an aggregate metric, like* r_3.

2.3.2 Variation point: SELECT

The variation point SELECT picks several chromosomes from a given set, and then retrieves their link rules. We define $\mathsf{SELECT}(C) = R$ where C is a set of chromosomes and R is a set of link rules. This function chooses from the set C a number of chromosomes following a certain heuristic, and then, retrieves their link rules. Some of the implementations may get the number of chromosomes to select as a configuration parameter; for instance this variation point may select five chromosomes but only output two link rules.

Example 2.8 *Assuming the set of chromosomes* $C = \{(r_1, 0.63), (r_2, 0.27), (r_3, 0.79)\}$, *this variation point may select the chromosomes* $(r_1, 0.63), (r_2, 0.27), (r_3, 0.79)$ *and output only the link rules* r_3 *and* r_2 *as a result. The heuristic to selected the link rules in this case is to choose the ones with the best and the worst evaluation scores.*

2.3.3 Variation point: CROSSOVER

The variation point CROSSOVER combines several link rules to produce new ones. We define $\mathsf{CROSSOVER}(R) = R'$ where R and R' are sets of

link rules. This function combines several link rules from R producing a new set of link rules R'.

Example 2.9 *Assuming as input* $R = \{r_2, r_3\}$ *where* r_2 *and* r_3 *were described in our previous examples. This variation point swaps the levenshtein in* r_2 *with the cosine in* r_3*. As a result* R' *is* $\{r'_4, r'_5\}$ *where:*

r'_4: $link(A, R)$ if $rdf{:}type(A) = dblp{:}Author$,
$rdf{:}type(R) = nsf{:}Researcher$,
$A_A = dblp{:}affiliation(A), N_R = nsf{:}name(R)$,
$S_1 = cosine(A_A, N_R)$,
$\frac{S_1 - 0.21}{1.00 - 0.21} > 0.$

r_5: $link(A, R)$ if $rdf{:}type(A) = dblp{:}Author$,
$rdf{:}type(R) = nsf{:}Researcher$,
$N_A = dblp{:}name(A), N_R = nsf{:}name(R)$,
$A_A = dblp{:}affiliation(A), U_R = nsf{:}university(R)$,
$S_1 = levenshtein(N_A, N_R), S_2 = jaccard(A_A, U_R)$,
$average(\frac{S_1 - 0.37}{1.00 - 0.37}, \frac{S_2 - 0.87}{1.00 - 0.87}) > 0.$

2.3.4 Variation point: MUTATE

The variation point MUTATE produces new link rules by modifying the input link rules. We define $MUTATE(R) = R'$ where R and R' are sets of link rules. This function chooses a rule from R, and then, modifies one or more elements in such rule depending on the implementation.

Example 2.10 *Assuming as input* $R = \{r'_4, r_5\}$ *from our previous example. This variation point may modify* r'_4 *by including a string transformation, i.e.,* $normalize$*. As a result the set* R' *contains* $\{r_4, r_5\}$*, where* r_4 *is:*

r_4: $link(A, R)$ if $rdf{:}type(A) = dblp{:}Author$,
$rdf{:}type(R) = nsf{:}Researcher$,
$A_A = dblp{:}affiliation(A), N_R = nsf{:}name(R)$,
$S_1 = cosine(A_A, normalize(N_R))$,
$\frac{S_1 - 0.21}{1.00 - 0.21} > 0.$

2.3.5 Variation point: REPLACE

The variation point REPLACE aims at combining two sets of chromosomes. We define $REPLACE(C, C') = C''$ where C, C' and C'' are sets of

chromosomes, such that C and C″ have the same size. This function replaces some of the chromosomes in C with the chromosomes in C′" producing as results C″, the heuristic used depends on the implementation.

Example 2.11 *Assuming as* C *the set* $\{(r_1, 0.63), (r_2, 0.27), (r_3, 0.79)\}$ *and as* C′ *the set* $\{(r_4, 1.00), (r_5, 0.94), (r_6, 0.63)\}$. *This variation point produces as result the set* C″ *containing* $\{(r_4, 1.00), (r_5, 0.94), (r_2, 0.27)\}$.

2.3.6 Variation point: STOP

The variation point STOP aims at establishing when a suitable solution has being found by the proposal, and thus, it should stop. We define $\text{STOP}(C) = t$ where C is a set of chromosomes and t is a Boolean value that specifies if a criterion was met. In addition this variation point requires two configuration parameters that are natural numbers, i.e., maximum iterations i and maximum generations g.

Example 2.12 *Assuming a value of* 3 *for the maximum iterations, a value of* 2 *as maximum generations, and* $\{(r_1, 0.63), (r_2, 0.27), (r_3, 0.79)\}$ *as the set* C. *When this variation point is called for the first time internally counts the current number iterations done, i.e., 1. Since this number is smaller than the maximum iterations, the variation point returns false. Then, this method is called again receiving* $\{(r_4, 1.00), (r_5, 0.94), (r_2, 0.27)\}$ *as the set* C, *internally counts the current iteration, i.e., 2. As a result, it outputs true since the solution has the maximum score, i.e.,* r_4 *with* 1.00.

2.3.7 Variation point: EVALUATE

The variation point EVALUATE aims at quantifying a rule relying on a confusion matrix; which is obtained by applying a certain link rule in the context of two datasets D_1 and D_2. We define $\text{EVALUATE}(M, r) = f$ where M is a confusion matrix, r the link rule related to such matrix, and f is a normalised number between 0.00 and 1.00 representing the effectiveness score achieved; where 0.00 is the worst achievable and is 1.00 the best.

Example 2.13 *Consider the confusion matrix* $M = (2, 1, 0, 0)$ *related to the link rule* r_1. *Then, the score of* r_1 *is* 0.66 *considering the Precision as function to obtain the effectiveness score; which is computed as* $tp/(tp + fp)$, *in this case* $2/(2 + 1)$.

	Carvalho	Eagle	Genlink
CREATE	Random-Trees	Random-Trees	Create-Genlink
SELECT	Roulette Wheel	Tournament	Tournament
REPLACE	Random	(μ + ß)	Generational
CROSSOVER	Tree-Crossover	Tree-Crossover	Crossover-Genlink
MUTATE	Tree-Mutation	Tree-Mutation	Mutation-Genlink
EVALUATE	F1	F1	F1

Table 2.1: *Variation points instantiations by literature proposals.*

	Gen1	Gen2	Gen3
CREATE	Random-Trees	Random-Trees	Random-Trees
SELECT	Tournament	Tournament	Roulette Wheel
REPLACE	(μ + ß)	(μ + ß)	Random
CROSSOVER	Tree-Crossover	Tree-Crossover	Crossover-Genlink
MUTATE	Mutation-Genlink	Tree-Mutation	Tree-Mutation
EVALUATE	F1	F1	F1

Table 2.2: *Variation points instantiations by our proposals.*

2.4 Implementations

We used our framework to implement three well-known supervised genetic programming-based proposals of link discovery from the literature, i.e., de Carvalho and others [21], Eagle [41], and Genlink [28].

To implement the proposals from the literature we instantiated the variations points CREATE, SELECT, CROSSOVER, MUTATE, REPLACE, STOP, and EVALUATE using the same functions as the original proposals. Table §2.1 describes the instantiation of these variation points depending on the proposal. Some of the functions presented in Table §2.1 are well-known in the context of the genetic programming [2, 38, 48], i.e., Random-Trees, Roulette Wheel, Tournament Selection, Random replace, $(\mu + \beta)$, Generational replace, Tree Crossover, Tree Mutation, and F_1. Nevertheless, others are custom functions that are not common in the genetic programming literature, i.e., Crossover Genlink, Mutation Genlink, and Objective Genlink.

The proposals in the literature can be categorised into elitist, i.e., Eagle, and random, i.e., Genlink and Carvalho. On the one hand, the elitist proposals prioritize the chromosomes that are more effective, i.e., have a higher

score. On the other hand, the random proposals prioritize chromosomes randomly or by mixing chromosomes with the highest and the lowest scores. The former approach is faster and very good in scenarios where the search space of solutions has few local maximums, as a drawback it is difficult for them to escape local maxima. The latter approach is slower since it explores more solutions from the search space, but it is unlikely that this approach will get stuck into a local maximum.

Considering the nature of the proposals, i.e., elitist or random approaches, and after analysing the proposals from the literature we present three new genetic programming proposals, i.e., Gen1, Gen2, and Gen3. The approach followed by Gen1 and Gen2 is highly elitist; on the contrary, Gen3 is highly random. Table §2.2 shows the instantiation of the variation points of our proposals, i.e., Gen1, Gen2, and Gen3.

In the following sub-sections, we aim at explaining the functions used in each of the variation points of our to template to implement the genetic programming algorithms from the literature, and our own.

2.4.1 CREATE implementations

The variation point CREATE that Carvalho, Eagle, Gen1, Gen2, and Gen3 rely on is known as Random-trees. Genlink implements a tailored-function for this variation point known as Create-Genlink. Next we provide a description for both:

Definition 2.6 (Random-trees) *This implementation randomly generates link rules, optionally, depending on the implementation this function can restrict the size of the link rules generated.*

Example 2.14 *This function may generate as link rule* r_1 *but it might also have generated a rule like* r_2*, both described previously. In this case, although* r_1 *is an acceptable rule,* r_2 *compares attributes that are not suitable in this scenario.*

Definition 2.7 (Create-genlink) *This function is a tailored-implementation proposed by Isele and Bizer [28]. Their implementation relies on two main steps, the former one aims at finding pairs of data properties, the second one to build link rules. The former applies the string metric levenshtein between all the values of the data properties within* D_1 *and* D_2*, those that obtain a score below a certain threshold* θ *are considered as a pair of suitable data properties. The later builds up a link rule that uses such properties: first*

it randomly selects an aggregation metric, second a random string metric is selected to compare one data property pair, third, with a probability of 0.50 *a transformation metric is added to one of the data properties. In addition, with a probability of* 0.50 *another comparison may be added to the aggregation following the same procedure.*

Example 2.15 *For instance, Create-Genlink may find as suitable data properties the following set of pairs {(*dblp:name, nsf:name*), (*dblp:affiliation, nsf:university*)} from the datasets DBLP and NSF described in Section §B.1. Considering one pair from such set of data property pairs, a first link rule is generated:*

$$\begin{aligned} r'\text{: } link(A,R) \text{ if } & rdf{:}type(A) = dblp{:}Author, \\ & rdf{:}type(R) = nsf{:}Researcher, \\ & N_A = dblp{:}name(A), N_R = nsf{:}name(R), \\ & S_1 = jaro(N_A, N_R), \\ & average(\frac{S_1 - 0.80}{1.00 - 0.80}) > 0. \end{aligned}$$

Then with a probability of 0.50 *per data property in* r' *a transformation metric may be appended. In our case let's assume that only the first data property in* r *obtained such probability; thus, the rule would now be* r''*:*

$$\begin{aligned} r''\text{: } link(A,R) \text{ if } & rdf{:}type(A) = dblp{:}Author, \\ & rdf{:}type(R) = nsf{:}Researcher, \\ & N_A = dblp{:}name(A), N_R = nsf{:}name(R), \\ & S_1 = jaro(capitalize(N_A), N_R), \\ & average(\frac{S_1 - 0.80}{1.00 - 0.80}) > 0. \end{aligned}$$

Finally, with a probability of 0.50 *another string metric may be added following the same procedure, in our case for the sake of this example let's assume that the final rule is* r''*.*

2.4.2 SELECT implementations

The variation point SELECT on which Carvalho and Genlink rely is known as Roulette Wheel. Eagle, Gen1, Gen2, and Gen3 rely on a function for this variation point known as Tournament. Next, we provide a description for both:

Definition 2.8 (Roulette Wheel) *This function is also known as Fitness proportionate selection. It assigns a fitness level* f_i *to each chromosome* c_i *in* C *following the next formula:*

$$f_i = \frac{c_i.v}{\sum_{i=1}^{|C|} score(c_i)}.$$

where c_i *is each of the chromosomes in* C *and* $score(c_i)$ *is the score assigned by the EVALUATE variation point to the link rule of such chromosome. Once all chromosomes have a fitness level assigned, a random number between* 0.00 *and* 1.00 *is generated and the chromosomes with such number as value of their fitness level are selected from* C*. Notice that the fitness levels are normalised, thus they fulfil that* $\sum_{i=1}^{|C|} f_i$ *has to be* 1.00*. The Roulette Wheel repeats this process until it has selected a number of link rules specified by a user.*

Example 2.16 *Consider that a user wishes to select tow chromosomes from set of chromosomes* $C = \{c_1, c_2, c_3\}$*, where* c_1 *is* $(r_1, 0.63)$*,* c_2 *is* $(r_2, 0.27)$*, and* c_3 *is* $(r_3, 0.79)$*. This implementation computes the following fitness levels:* 0.37 *for* c_1*,* 0.16 *for* c_2*, and* 0.47 *for* c_3*. Now a random number is generated, for instance* 0.32*. Since* 0.32 *is between the fitness level of* c_1*, i.e.,* 0.37*, and* c_2*, i.e.,* 0.16*, then the chromosome* c_1 *is selected. Next, the second chromosome is selected, this time the random number is* 0.70*; since* 0.70 *is above the fitness level of* c_3*, i.e.,* 0.47*, then this chromosome is selected. As a result the set of link rules* $R = \{r_1, r_3\}$ *is output.*

Definition 2.9 (Tournament) *This function receives from a user two additional configuration parameters, a tournament size* t_s *and the number of link rules to be selected. The function takes a number of* t_s *chromosomes from* C *randomly, then from those chosen, it keeps the one with the best score by means of EVALUATE. Finally, this function returns the link rule of the selected chromosome. This process is repeated until the number of link rules specified by the user is selected.*

Example 2.17 *Consider that a user wishes to select three chromosomes, the tournament size* t_s *is 2, and the set of chromosomes* $C = \{c_1, c_2, c_2\}$*, where* c_1 *is* $(r_1, 0.63), c_2$ *is* $(r_2, 0.27)$*, and* c_3 *is* $(r_3, 0.79)$*. This implementation randomly selects two chromosomes from* C*, for instance* c_1 *and* c_3*. Then it keeps the one with the best score, i.e.,* c_3*, and extracts its rule* r_3*. Next, the second rule is selected using the same procedure, assuming this time* c_1 *and* c_2 *were selected, the chosen link rule would be* r_1 *since its chromosome* c_1 *has a score of* 0.63 *which is the highest. Finally, the third link rule is selected similarly; assuming that this time* c_3 *and* c_1 *were selected the output rule would be* r_3*. As a result, the set* $R = \{r_1, r_3\}$ *of link rules is output.*

2.4.3 CROSSOVER implementations

The variation point CROSSOVER that Carvalho, Eagle, Gen1, and Gen2 rely on is known as Tree-Crossover. Genlink and Gen3 rely on a tailored-function for this variation point known as Crossover-Genlink. Next, we provide a description for both:

Definition 2.10 (Tree-Crossover) *This function is meant to receive two link rules. Then it selects in both two compatible elements, which can be either metrics, attribute layers, or thresholds. By compatible we refer to metrics of the same kind, for instance an aggregate metric is not compatible with a string metric. Next, this function swaps the selected elements, along with, their nested elements. As a result, two new link rules are output.*

Example 2.18 *Consider as input the link rules* r_1 *and* r3 *from our examples. First, two pair of compatible elements are selected, for instance, if the selected element in* r3 *is* average *then no crossover can be applied since* r_1 *has no compatible elements, i.e., aggregate metrics. Assuming* jaro *is selected in* r_1*, the available options in* r3 *are* cosine *and* jaccard*, the resulting rule of swapping with cosine would be* r_5*. Assuming* 0.92 *is selected from* r_1 *and* 0.87 *from* r_3*, then the result would be the following because the thresholds are the only compatible elements to swap in both cases:*

$$
\begin{aligned}
r_7\text{: } \mathtt{link}(A,R) \text{ if } & \mathtt{rdf{:}type}(A) = \mathtt{dblp{:}Author},\\
& \mathtt{rdf{:}type}(R) = \mathtt{nsf{:}Researcher},\\
& N_A = \mathtt{dblp{:}name}(A), N_R = \mathtt{nsf{:}name}(R),\\
& S_1 = \mathtt{jaro}(N_A, N_R),\\
& \frac{S_1 - 0.87}{1.00 - 0.87} > 0.
\end{aligned}
$$

$$
\begin{aligned}
r_8\text{: } \mathtt{link}(A,R) \text{ if } & \mathtt{rdf{:}type}(A) = \mathtt{dblp{:}Author},\\
& \mathtt{rdf{:}type}(R) = \mathtt{nsf{:}Researcher},\\
& N_A = \mathtt{dblp{:}name}(A), N_R = \mathtt{nsf{:}name}(R),\\
& A_A = \mathtt{dblp{:}affiliation}(A), U_R = \mathtt{nsf{:}university}(R),\\
& S_1 = \mathtt{cosine}(N_A, N_R), S_2 = \mathtt{jaccard}(A_A, U_R),\\
& \mathtt{average}(\frac{S_1 - 0.37}{1.00 - 0.37}, \frac{S_2 - 0.92}{1.00 - 0.92}) > 0.
\end{aligned}
$$

Definition 2.11 (Crossover-Genlink) *This is a tailored-function devised by Isele and Bizer [28], which is meant to be applied between two link rules. This function randomly selects one element from a link rule, and then randomly applies one sub-crossover function from the list below. The selected*

sub-crossover depends on the selected element, since not all are applicable to all the elements. The available sub-crossover functions are:

- *Tree-crossover: this function is the same as the one explained above in the Tree-Crossover.*
- *Mixed-crossover: when a string metric is chosen from a former link rule and an aggregate metric is chosen from a second link rule to be crossed, this function changes the first string metric with the aggregate metric.*
- *Aggregate-crossover: when an aggregate metric is chosen to be crossed, this function randomly selects in both input rules an aggregate metric. Then, it appends to the first all the string metrics from the second. Next, for each element appended in the first aggregation it removes with a probability of* 0.50 *each of them.*
- *Transformation-crossover: when a transformation is chosen to be crossed, this function replaces in the former rule the selected transformation with the transformation selected in the second link rule; if the second transformation has nested transformation metrics these are also included in the first rule.*
- *Threshold-crossover: when a threshold is chosen to be crossed, this function computes the average of two selected thresholds and sets as new threshold in the first rule such value.*
- *Constant-Crossover: when a constant is chosen to be crossed, this function does the same operations performed by the threshold-crossover.*

Example 2.19 (Mixed-crossover) *Assuming the rules* r_1 *and* r_3 *from our examples. Then, the selected string metric in the former rule could be jaro and the aggregate metric in the later is average, then this function replaces the former string metric with the later aggregate metric, and then appends the former string metric. As a result the new output link rule would be:*

$$
\begin{aligned}
r_9\text{: } \mathrm{link}(A,R) \text{ if } & \mathrm{rdf{:}type}(A) = \mathrm{dblp{:}Author},\\
& \mathrm{rdf{:}type}(R) = \mathrm{nsf{:}Researcher},\\
& N_A = \mathrm{dblp{:}name}(A), N_R = \mathrm{nsf{:}name}(R),\\
& S_1 = \mathrm{jaro}(N_A, N_R),\\
& \mathrm{average}(\frac{S_1 - 0.92}{1.00 - 0.92}) > 0.
\end{aligned}
$$

Example 2.20 (Aggregate Crossover) *Assuming the rules* r_3 *and* r_9 *from our examples. Then, this function randomly selects one aggregate metric from both rules, i.e., average in this case because it is the only aggregate metric that they have. Following, it appends to the first link rule all the string metrics of the second:*

$$
\begin{aligned}
r'_{10}\text{: } \mathrm{link}(A,R) \text{ if } & \mathrm{rdf{:}type}(A) = \mathrm{dblp{:}Author},\\
& \mathrm{rdf{:}type}(R) = \mathrm{nsf{:}Researcher},\\
& N_A = \mathrm{dblp{:}name}(A), N_R = \mathrm{nsf{:}name}(R),\\
& A_A = \mathrm{dblp{:}affiliation}(A), U_R = \mathrm{nsf{:}university}(R),\\
& S_1 = \mathrm{jaro}(N_A, N_R),\\
& S_2 = \mathrm{cosine}(N_A, N_R),\\
& S_3 = \mathrm{jaccard}(A_A, U_R),\\
& \mathrm{average}(\frac{S_1 - 0.92}{1.00 - 0.92}, \frac{S_2 - 0.37}{1.00 - 0.37}, \frac{S_3 - 0.87}{1.00 - 0.87}) > 0.
\end{aligned}
$$

Next, for each of the elements append to average a probability is obtained, e.g., 0.60 *for the* jaro*,* 0.93 *for the* cosine*, and* 0.23 *for* jaccard*. Then elements with a probability below* 0.50 *are removed. As a result the new rule is:*

$$
\begin{aligned}
r_{10}\text{: } \mathrm{link}(A,R) \text{ if } & \mathrm{rdf{:}type}(A) = \mathrm{dblp{:}Author},\\
& \mathrm{rdf{:}type}(R) = \mathrm{nsf{:}Researcher},\\
& N_A = \mathrm{dblp{:}name}(A), N_R = \mathrm{nsf{:}name}(R),\\
& S_1 = \mathrm{jaro}(N_A, N_R),\\
& S_2 = \mathrm{cosine}(N_A, N_R),\\
& \mathrm{average}(\frac{S_1 - 0.92}{1.00 - 0.92}, \frac{S_2 - 0.37}{1.00 - 0.37}) > 0.
\end{aligned}
$$

Example 2.21 (Transformation-crossover) *Considering the following new rules:*

$$
\begin{aligned}
r'_{11}\text{: } \mathrm{link}(A,R) \text{ if } & \mathrm{rdf{:}type}(A) = \mathrm{dblp{:}Author},\\
& \mathrm{rdf{:}type}(R) = \mathrm{nsf{:}Researcher},\\
& N_A = \mathrm{dblp{:}name}(A), N_R = \mathrm{nsf{:}name}(R),\\
& S_1 = \mathrm{jaro}(\mathrm{uppercase}(N_A), N_R),\\
& \frac{S_1 - 0.70}{1.00 - 0.70} > 0.
\end{aligned}
$$

$$
\begin{aligned}
r_{12}\text{: } \mathrm{link}(A,R) \text{ if } & \mathrm{rdf{:}type}(A) = \mathrm{dblp{:}Author},\\
& \mathrm{rdf{:}type}(R) = \mathrm{nsf{:}Researcher},\\
& N_A = \mathrm{dblp{:}name}(A), N_R = \mathrm{nsf{:}name}(R),\\
& S_1 = \mathrm{jaccard}(N_A, \mathrm{tokenise}(\mathrm{stem}(\mathrm{lowercase}(N_R)))),\\
& \frac{S_1 - 0.19}{1.00 - 0.19} > 0.
\end{aligned}
$$

This function selects in both link rules a transformation metric, e.g., uppercase from the former and stem from the later. Then it replaces the metric in the first link rule with the one in the second, and its nested transformation metrics. As a result, the rule obtained is:

$$\begin{aligned} r_{11}\text{: } & \mathtt{link}(A,R) \text{ if } \mathtt{rdf{:}type}(A) = \mathtt{dblp{:}Author},\\ & \mathtt{rdf{:}type}(R) = \mathtt{nsf{:}Researcher},\\ & N_A = \mathtt{dblp{:}name}(A), N_R = \mathtt{nsf{:}name}(R),\\ & S_1 = \mathtt{jaccard}(\mathtt{stem}(\mathtt{lowercase}(N_A)), \mathtt{lowercase}(N_R)),\\ & \frac{S_1 - 0.70}{1.00 - 0.70} > 0. \end{aligned}$$

Example 2.22 (Threshold-crossover) *Assuming the rules r_1 and r_2 from our examples. Then, this function first selects randomly two thresholds, one from each rule, e.g., 0.92 from r_1 and 0.21 from r_2. Then, it computes their average value and replaces the threshold in the former link rule, i.e., 0.57 in r_1 instead of 0.92. As a result, the new rule would be:*

$$\begin{aligned} r_{13}\text{: } & \mathtt{link}(A,R) \text{ if } \mathtt{rdf{:}type}(A) = \mathtt{dblp{:}Author},\\ & \mathtt{rdf{:}type}(R) = \mathtt{nsf{:}Researcher},\\ & N_A = \mathtt{dblp{:}name}(A), N_R = \mathtt{nsf{:}name}(R),\\ & S_1 = \mathtt{jaro}(N_A, N_R),\\ & \frac{S_1 - 0.57}{1.00 - 0.57} > 0. \end{aligned}$$

Example 2.23 (Constant-Crossover) *Considering the rules:*

$$\begin{aligned} r'_{14}\text{: } & \mathtt{link}(A,R) \text{ if } \mathtt{rdf{:}type}(A) = \mathtt{dblp{:}Author},\\ & \mathtt{rdf{:}type}(R) = \mathtt{nsf{:}Researcher},\\ & N_A = \mathtt{dblp{:}name}(A), N_R = \mathtt{nsf{:}name}(R),\\ & S_1 = \mathtt{jaro}(N_A, N_R),\\ & \mathtt{multiply}(\frac{S_1 - 0.13}{1.00 - 0.13}, 0.93) > 0. \end{aligned}$$

$$\begin{aligned} r_{15}\text{: } & \mathtt{link}(A,R) \text{ if } \mathtt{rdf{:}type}(A) = \mathtt{dblp{:}Author},\\ & \mathtt{rdf{:}type}(R) = \mathtt{nsf{:}Researcher},\\ & A_A = \mathtt{dblp{:}affiliation}(A), N_R = \mathtt{nsf{:}name}(R),\\ & S_1 = \mathtt{cosine}(A_A, N_R),\\ & \mathtt{minimum}(\frac{S_1 - 0.57}{1.00 - 0.57}, 0.29) > 0. \end{aligned}$$

This function first selects randomly two constants, one from each rule, e.g., 0.93 from r_{14} and 0.29 from r_{15}. Then, it computes their average value,

i.e., 0.61*. The new value replaces the constant selected in the former rule. As a result, the new rule is:*

$$\begin{aligned} r_{14}\text{: } link(A,R) \text{ if } & rdf{:}type(A) = dblp{:}Author, \\ & rdf{:}type(R) = nsf{:}Researcher, \\ & N_A = dblp{:}name(A), N_R = nsf{:}name(R), \\ & S_1 = jaro(N_A, N_R), \\ & multiply(\frac{S_1 - 0.13}{1.00 - 0.13}, 0.61) > 0. \end{aligned}$$

2.4.4 MUTATE implementations

The variation point MUTATE that Carvalho, Eagle, Gen1, and Gen2 rely on is known as Tree-Mutation. Genlink and Gen3 rely on a tailored-function for this variation point known as Mutation-Genlink. Next, we provide a description for both:

Definition 2.12 (Tree-Mutation) *This function relies on a pool of available metrics of different types, i.e., aggregate, string, and transformation, and two sets of data properties from two datasets each. It selects one element from a link rule and randomly changes it from another that belongs to the pool of metrics (in this case puts one compatible metric), data property pair, or a random number if the selected element was a threshold or a constant. This process is repeated for a number of link rules as specified by a user.*

Example 2.24 *Consider the rule* r_{14} *from our examples, and the set of metrics* $\{maximum, average, minimum, levenshtein, jaro\}$ *and the set of data properties {*dblp:name, dblp:affiliation*} for DBLP and {*nsf:name, nsf:university*} NSF datasets; respectively. Assuming that the selected element from* r_{14} *is* multiply*, then a new rule* r_{16} *could replace the aggregate metric from the another list, i.e.,* average*. Assuming that the selected element is* jaro *then a new rule* r_{17} *could replace such metric from another form the list, i.e.,* cosine*. Assuming the selected element is* dblp:name*, then the new rule* r_{18} *could replace such property from any other contained the list of DBLP data properties, i.e.,* dblp:affiliation*. Assuming a threshold or a constant is selected form* r_{14} *it will be substitute by a random number, i.e.,* r_{19} *with the new threshold 0.85 or* r_{20} *with the constant replaced by* 0.08*. In all these cases the output rules would be:*

$$\begin{aligned} r_{16}\text{: } link(A,R) \text{ if } & rdf{:}type(A) = dblp{:}Author, \\ & rdf{:}type(R) = nsf{:}Researcher, \\ & N_A = dblp{:}name(A), N_R = nsf{:}name(R), \end{aligned}$$

$$S_1 = \mathrm{jaro}(N_A, N_R)$$
$$\mathrm{average}(\frac{S_1 - 0.13}{1.00 - 0.13}, 0.61) > 0.$$

r_{17}: $\mathrm{link}(A, R)$ if $\mathrm{rdf{:}type}(A) = \mathrm{dblp{:}Author}$,
$\mathrm{rdf{:}type}(R) = \mathrm{nsf{:}Researcher}$,
$N_A = \mathrm{dblp{:}name}(A), N_R = \mathrm{nsf{:}name}(R)$,
$S_1 = \mathrm{cosine}(N_A, N_R)$
$\mathrm{multiply}(\frac{S_1 - 0.13}{1.00 - 0.13}, 0.61) > 0.$

r_{18}: $\mathrm{link}(A, R)$ if $\mathrm{rdf{:}type}(A) = \mathrm{dblp{:}Author}$,
$\mathrm{rdf{:}type}(R) = \mathrm{nsf{:}Researcher}$,
$A_A = \mathrm{dblp{:}affiliation}(A), N_R = \mathrm{nsf{:}name}(R)$,
$S_1 = \mathrm{jaro}(A_A, N_R)$
$\mathrm{multiply}(\frac{S_1 - 0.13}{1.00 - 0.13}, 0.61) > 0.$

r_{19}: $\mathrm{link}(A, R)$ if $\mathrm{rdf{:}type}(A) = \mathrm{dblp{:}Author}$,
$\mathrm{rdf{:}type}(R) = \mathrm{nsf{:}Researcher}$,
$N_A = \mathrm{dblp{:}name}(A), N_R = \mathrm{nsf{:}name}(R)$,
$S_1 = \mathrm{jaro}(N_A, N_R)$
$\mathrm{multiply}(\frac{S_1 - 0.85}{1.00 - 0.85}, 0.61) > 0.$

r_{20}: $\mathrm{link}(A, R)$ if $\mathrm{rdf{:}type}(A) = \mathrm{dblp{:}Author}$,
$\mathrm{rdf{:}type}(R) = \mathrm{nsf{:}Researcher}$,
$N_A = \mathrm{dblp{:}name}(A), N_R = \mathrm{nsf{:}name}(R)$,
$S_1 = \mathrm{jaro}(N_A, N_R)$
$\mathrm{multiply}(\frac{S_1 - 0.13}{1.00 - 0.13}, 0.08) > 0.$

Definition 2.13 (Mutate-Genlink) *This is a tailored-function devised by Isele and Bizer [28], which receives a singleton set of link rules* R*. It first creates a random link rule by means of Create-Genlink, and then, applies a Crossover-Genlink function between the new link rule and the one in the input set* R*. The new link rule is output within a unary set of link rules* R'*.*

Example 2.25 *Consider the rule* r_{14} *from our examples as input. Then, Mutate-Genlink first creates a new link rule relying on Create-Genlink, for instance:*

r'_{21}: $\mathrm{link}(A, R)$ if $\mathrm{rdf{:}type}(A) = \mathrm{dblp{:}Author}$,

$$\begin{aligned}&\text{rdf:type}(R) = \text{nsf:Researcher},\\ &N_A = \text{dblp:name}(A), N_R = \text{nsf:name}(R),\\ &S_1 = \text{jaro}(\text{uppercase}(N_A), N_R)\\ &\text{average}(\frac{S_1 - 0.43}{1.00 - 0.43}) > 0.\end{aligned}$$

Then, it applies the Crossover-Genlink to obtain a new link rule. As a result, the new mutated rule could be:

$$\begin{aligned}r_{21}\text{: link}(A, R) \text{ if } &\text{rdf:type}(A) = \text{dblp:Author},\\ &\text{rdf:type}(R) = \text{nsf:Researcher},\\ &N_A = \text{dblp:name}(A), N_R = \text{nsf:name}(R),\\ &S_1 = \text{jaro}(\text{uppercase}(N_A), N_R)\\ &S_2 = \text{jaro}(N_A, N_R)\\ &\text{average}(\frac{S_1 - 0.43}{1.00 - 0.43}, \frac{S_2 - 0.13}{1.00 - 0.13}) > 0.\end{aligned}$$

2.4.5 REPLACE implementations

The function for the variation point REPLACE in which Carvalho relies is known as Random. Eagle, Gen1, and Gen2 rely on a function known as $(\mu + \beta)$. Genlink and Gen3 rely on a function known as Generational. Next, we provide a description for all these functions:

Definition 2.14 (Random) *This function randomly selects chromosomes from C and C' and stores them in the set C'' until the size of this set equals the size of C.*

Example 2.26 *Consider the set of chromosomes $C = \{(r_1, 0.63), (r_2, 0.27), (r_3, 0.79)\}$ and $C' = \{(r_4, 1.00), (r_5, 0.94), (r_6, 0.63)\}$. This function may generate as resulting set $C'' = \{(r_2, 0.27), (r_4, 0.80), (r_5, 0.10)\}$, by selecting the first chromosome from the set C and the other two from C'.*

Definition 2.15 ($(\mu + \beta)$) *This function combines the input set of chromosomes C and C' into a new one, i.e., C''. Then, it sorts the chromosomes in C'' by the score of their rules, and finally keeps a number of $|C|$ chromosomes that have the highest scores.*

Example 2.27 *Consider the set of chromosomes $C = \{(r_1, 0.63), (r_2, 0.27), (r_3, 0.79)\}$ and $C' = \{(r_4, 1.00), (r_5, 0.94), (r_6, 0.63)\}$. This function creates a resulting set that combines both C and C' chromosomes, i.e., $C'' = \{(r_1, 0.63), (r_2, 0.27), (r_3, 0.79), (r_4, 1.00), (r_5, 0.94), (r_6, 0.63)\}$. Then this*

function sorts the chromosomes in C'' *by the score of the link rules, i.e.,* $C'' = \{(r_4, 1.00), (r_5, 0.94), (r_3, 0.79), (r_1, 0.63), (r_6, 0.63), (r_2, 0.27)\}$. *Finally, it keeps only the first* $|C|$ *chromosomes in* C'' *with the highest score. As a result, the set* $C'' = \{(r_4, 1.00), (r_5, 0.94), (r_3, 0.79)\}$ *is output.*

Definition 2.16 (Generational) *This function favours the set of chromosomes representing the offspring. Therefore, this function always returns the set of chromosomes corresponding to them.*

Example 2.28 *Consider the set of chromosomes* $C = \{(r_1, 0.63), (r_2, 0.27), (r_3, 0.79)\}$ *and* $C' = \{(r_4, 1.00), (r_5, 0.94), (r_6, 0.63)\}$*; being the former set the parents, and the later the offspring set. This function returns a new set of chromosomes* C'.

2.4.6 STOP implementation

The STOP variation point requires a set of chromosomes C, a number of maximum iterations, i, and a maximum of generations g. In the literature, as far as we know, no other implementation rather than the standard one has being used by proposals to generate link rules.

Definition 2.17 (Standard STOP) *This implementation relies on the following criterion: a) Each time* STOP *is called internally counts the iterations done, when this number is equal to* i *the variation point* STOP *outputs true; b) If during a number of* g *iterations the fitness score of all the chromosomes in* C *does not changed at all the variation point* STOP *outputs true; c) When a chromosome has a fitness score of 1.00, i.e. the maximum possible, the variation point* STOP *outputs true.*

Example 2.29 *Consider a value of 2 for the maximum iterations, and* $\{(r_{16}, 0.84), (r_{17}, 0.76), (r_{18}, 0.44)\}$ *as the set* C. *When this variation point is called for the first time internally counts as current iterations 1. Since this number is smaller than the maximum iterations, the variation point returns false. Then, this method is called again receiving* $\{(r_{17}, 0.84), (r_{18}, 0.44), (r_{20}, 0.62)\}$ *as the set* C*, internally counts as current iterations 2. As a result, it outputs true since the current iterations have reached the same value of the maximum iterations.*

2.4.7 EVALUATE implementations

The EVALUATE variation point requires a function to compute an effectiveness score f relying on a confusion matrix, i.e., $M = (tp, fp, tn, fn)$, and

its related link rule. Carvalho, Eagle, Gen1, Gen2, and Gen3 rely on the F_1, and Genlink on a tailored-version of the Matthews Correlation Coefficient known as Objective-Genlink. Next, we explain both formulas:

Definition 2.18 (F_1) *This function computes the effectiveness score as follows* $f = 2 * P * R/(P + R)$*, where* $P = tp/(tp + fp)$ *and* $R = tp/(tp + fn)$*. As a result, a normalised score between* 0.00 *and* 1.00 *is output; where* 0.00 *is the lowest and* 1.00 *the highest achievable.*

Definition 2.19 (EVALUATE-Genlink) *This function is based on the Matthews Correlation Coefficient formula mcc. It combines the result of such formula with the number of metrics that a link rule has, relying on the function* op*. This function is meant to penalize the effectiveness of a link rule in base of how many functions it has. This function is defined as follows* $f = mcc - 0.05 * op(r)$*, where* mmc *is* $((tp * tn) - (fp * fn))/\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}$*.*

2.5 Experimental analysis

In this section, we introduce the results and conclusions achieved in our experiments. We aim at executing the proposals form the literature, and ours, under the same experimental environment. Our goal is to determine the proposal that behaves the best considering five well-known datasets from the literature.

2.5.1 Experimental environment

Implementation. We implemented our framework[†1] using Java 1.8 and the following components: Jena TDB 3.2.0 to work with RDF data, Jena ARQ 3.2.0 to work with SPARQL queries, MOEA 2.12 framework to work with genetic techniques, and Simmetrics 1.6.2, Secondstring, and JavaStringSimilarity to work with string similarity functions. We used the implementations provided by the original papers for functions CREATE, SELECT, CROSSOVER, MUTATE, REPLACE, EVALUATE, and OBJECTIVE.

Running experiments. We run our experiments in the facility described in Appendix §A.1. In addition we defined 49 different setups to run each proposal, each setup is a tuple (i, p, p_c, p_m, g); where i is the maximum number of iterations, p is the maximum population size, p_c is the crossover probability, p_m is the mutation probability, and g is the maximum number of generations. Appendix §A.3 reports all the setup specified to run our experiments.

[†1]The framework is available at https://github.com/AndreaCimminoArriaga/EvA4LD.

Running method. We run the proposals following a 2-fold cross validation, for each setup defined we executed ten times each proposal; as suggested by Isele and Bizer [28] in their paper on Genlink. Carvalho and Eagle articles did not provide detailed information of how they ran their experiments. In addition, Isele and Bizer [28] compared Genlink and Carvalho proposals in their article; therefore, we decided that their running method was the most suitable to follow.

Datasets sizes. Following the experimental methodology defined by Isele and Bizer [28] in their paper on Genlink, we split the datasets of the scenarios into other two sub-datasets of smaller sizes with non-overlapping resources; in addition we provided a sub-set of the links owl:sameAs and owl:differentFrom existing between the resources in such sub-datasets. Relying on the scenarios described in Appendix §A.2, we defined the size of the sub-datasets according to the ones used by Isele and Bizer [28]:

- In the Restaurants and RestaurantsZ scenarios we used 112 resources per sub-dataset, from which 56 are related by means of owl:sameAs and the rest by owl:differentFrom.
- In the Persons1 scenario we used 500 resources per sub-dataset, from which 250 resources are related by means of owl:sameAs, and the rest by means of owl:differentFrom.
- In scenario Persons2 each sub-dataset has 400 resources, from which 200 are related by means of owl:sameAs and the rest by owl:differentFrom
- In the scenario Articles each sub-dataset counts with 1600 resources from which 800 are related by means of owl:sameAs, and the rest by means of owl:differentFrom.

2.5.2 Experimental results

In this section, we introduce the results of our experimentation. Figures §2.3, §2.4, §2.5, §2.6, and §2.7 depict the precision versus recall obtained by the resultant link rules when they were applied in the validation phase. Our goal is to check how the link rules generated by the proposals behave in our scenarios.

In Figures §2.3, §2.4, §2.5, §2.6, and §2.7 it can be observed that proposals generate link rules that behave well in all the scenarios. In general

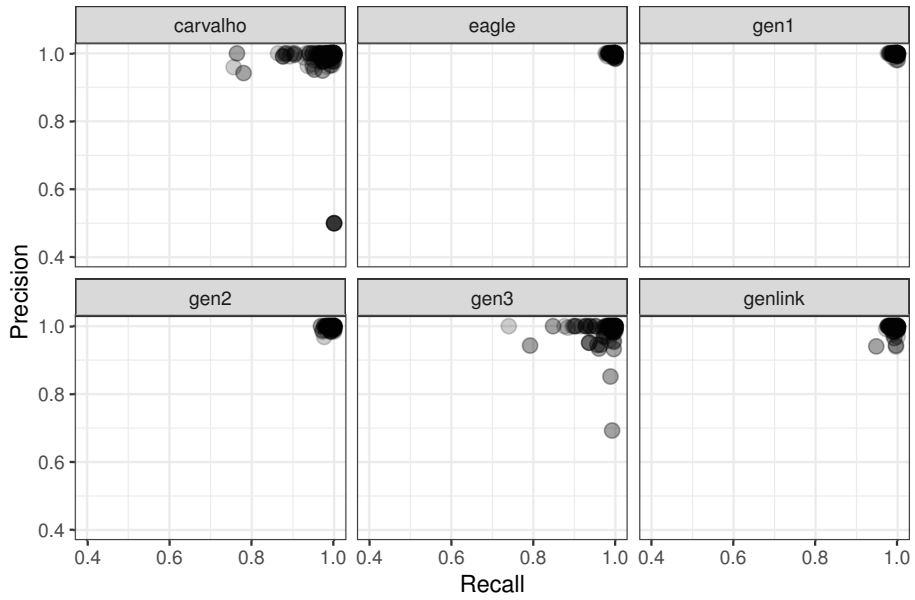


Figure 2.3: *Evaluation results in scenario Persons1.*

terms Gen1, Gen2, and Eagle always obtain results that seem to be better, and be grouped in the same place. Carvalho, Genlink, and Gen3 tend to obtain sparser results, from which some are not as good as the results obtained by the previous proposals. We can conclude that there are two patterns of behaviour, some proposals generate link rules focused in the same solution space, and other, obtain sparser results. However, we still cannot conclude which proposal obtained better results in these scenarios.

2.5.3 Statistical analysis

The results obtained by the different proposals in each scenario are shown in Table §2.3, which expose the average validation value obtained by the different proposals with all their setups. The first column of Table §2.3 refers to the scenarios, and the rest of columns are the average F_1 achieved by each of the proposals in such scenario. We highlighted the proposals that obtained the highest F_1 in a given scenario. In addition, we run a Bergmann-Hommel's ranking (using a p-value of 0.05) to determine in this scenarios which proposal performs better, Table §2.4 reports the ranking results. Notice that the ranking only reports the best proposals in the context of these scenarios, not as a global result.

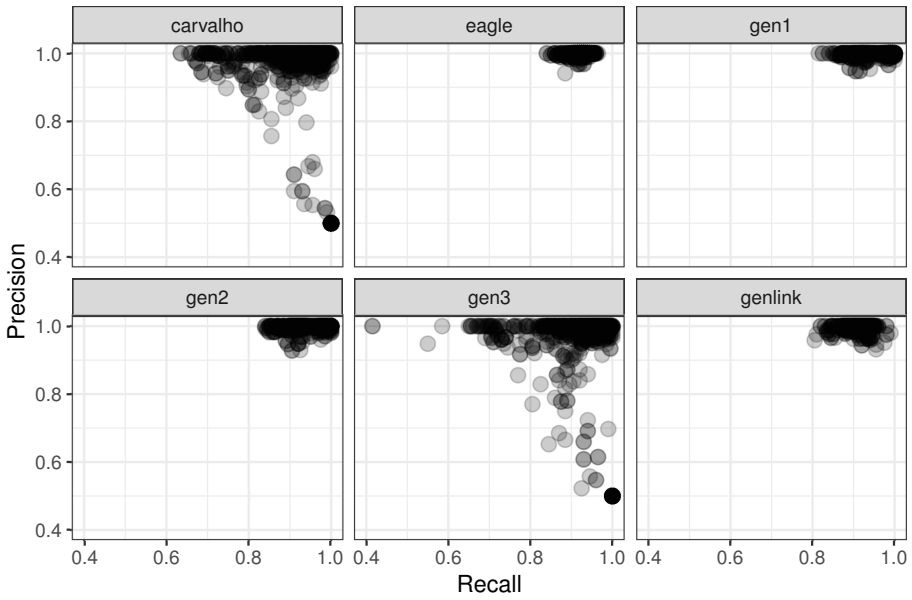


Figure 2.4: *Evaluation results in scenario Persons2.*

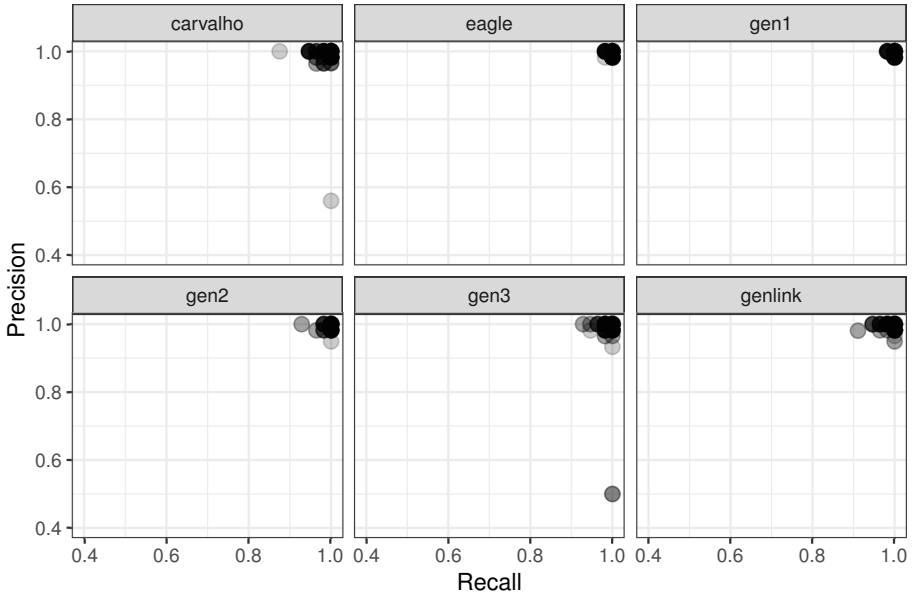


Figure 2.5: *Evaluation results in scenario Restaurants.*

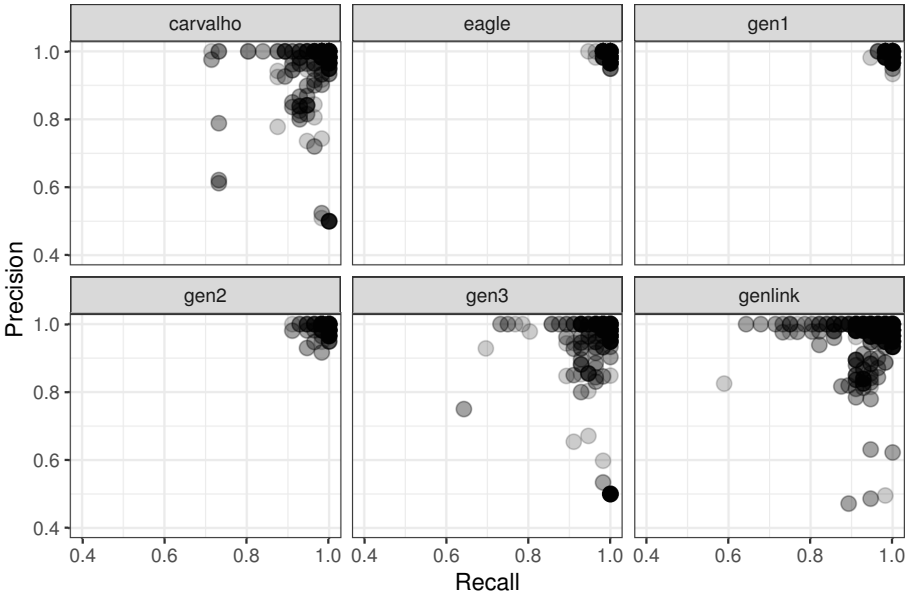


Figure 2.6: *Evaluation results in scenario RestaurantsZ.*

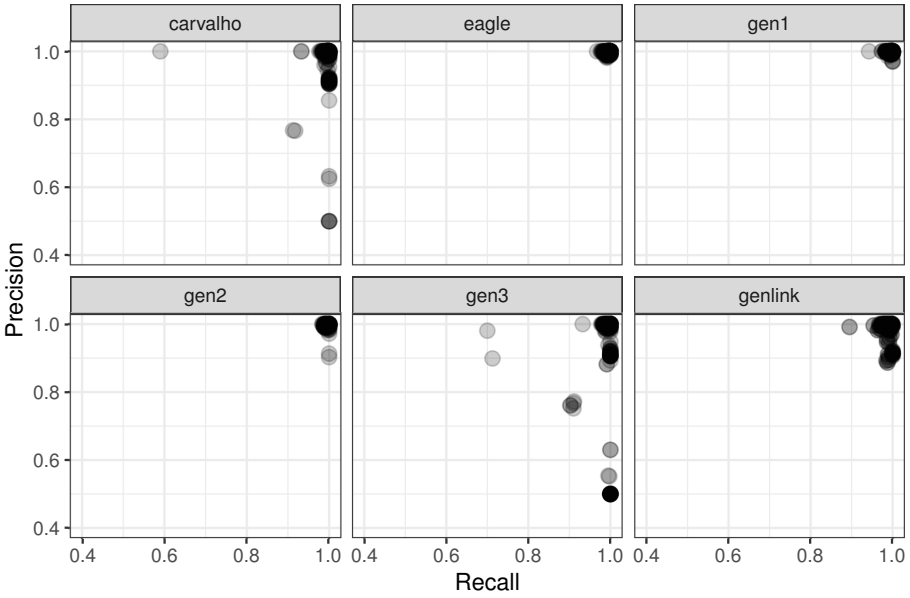


Figure 2.7: *Evaluation results in scenario Articles.*

	Carvalho	Eagle	Gen1	Gen2	Gen3	Genlink
Persons1	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Persons2	0.90	0.96	**0.98**	**0.98**	0.90	0.95
Articles	0.99	**1.00**	**1.00**	**1.00**	0.99	0.99
Restaurants	0.99	**1.00**	**1.00**	**1.00**	0.99	0.98
RestaurantsZ	0.98	**1.00**	**1.00**	**1.00**	0.97	0.97

Table 2.3: *Average* F_1 *score.*

	Persons1	Persons2	Articles	Restaurants	RestaurantsZ
#1	Gen1, Gen2, Eagle, Genlink	Gen1, Gen2	Gen1	Gen1, Gen2	Gen1, Gen2, Eagle
#2	Gen3	Eagle	Gen2	Carvalho, Eagle, Gen3	Carvalho, Gen3
#3	Carvalho	Genlink	Carvalho, Eagle	Genlink	Genlink
#4	-	Carvalho, Gen3	Gen3	-	-
#5	-	-	Genlink	-	-

Table 2.4: *Bergmann-Hommel's ranking based on* F_1 *score.*

Observing Table §2.3 the proposals Gen1 and Gen2 always achieve the highest F_1 independently of the scenario, Gen1 achieves as well the highest F_1 except in the Persons2 scenario. Relying on this results we could conclude that Gen1 and Gen2 are the proposals that perform the best; however this would not be true since, on the one hand we do not know which perform better between Gen1 and Gen2, and, on the other hand, we do not have statistical support to state that they perform better that the others in terms of effectiveness.

In order to determine, in the context of these scenarios, the proposal that performs better we ran the Bergmann-Hommel's ranking, which results are exposed in Table §2.4. Relying on this table we observe that Gen1 and Gen2 are always in the first place of the rank, but sometimes we cannot affirm that they are better than the rest since, for instance, in Persons1 the proposal Genlink is in the position #1 of the ranking as well. As a result, we conclude that Gen1 and Gen2 behave better than the rest, although in some specific scenarios like Persons1 they may be as good as other proposals that usually rank in the last position most of the times.

2.6 Summary

In this chapter, we have introduced a framework that we have devised, which provides a harness to implement a variety of genetic programming-based link discovery proposals. Our framework relies on a template with several variation points that, once implemented by specific functions, conforms a genetic programming-based proposal. Relying in the framework we have implemented three proposals from the literature, and we presented three custom proposals. Finally, we have presented the results of applying the proposals in five different scenarios by ranking their results using a Bergmann-Hommel test.

Chapter 3

Teide: Bootstrapping Link Rules

This chapter introduces our proposal to bootstrap link rules, which has been proved that increases the precision of the rules without a significant drop in their recall. It is organised as follows: Section §3.1 introduces the context of our work; Section §3.2 provides the details of our proposal; Section §3.3 describes the experiments conducted with Teide; and finally, Section §3.4 recaps on the conclusions drawn from our experiments.

3.1 Introduction

In the previous chapter, we introduced the framework that we have devised to implement genetic programming-based proposals for link discovery. These proposals learn link rules that are used to link resources that refer to the same real-world entities but that are allocated in two different datasets. However, the rules learnt rely only on the data properties of the resources that are analysed to be linked. We found out that these rules have a drawback when applied. These rules are not able to cope with resources with very similar data properties that refer to different entities, entailing that their precision drops in this kind of scenarios.

We devised a new proposal known as Teide that applies link rules following a different approach, which is able to cope with resources that have similar data properties. Teide consists in three main components, namely: the first one learns link rules relying on any proposal from the literature, the second one filters out the links that the link rules produce when applied, and the third relies on two voting strategies to select the most reliable filtered links.

The link rule learner can be any implementation built with our framework from the previous chapter. It requires a set of links owl:sameAs and owl:differentFrom in order to learn a set of link rules. The implementation should learn link rules for different types of resources in the datasets, from which at least one learnt rules one should be kept. As a result, we obtain a list of link rules that link different types of resources, from this set one is selected to be improved using our approach and the rest are the supporting rules. Notice that as far as we obtain a list of link rules to be applied between different types of resources, the method to obtain such rules does not matter.

The filter is an ad-hoc component that works as follows: it takes a link rule and executes it to produce a set of candidate links; then, it analyses the neighbours of the resources involved in each candidate link by boosting the remaining rules; links in which the corresponding neighbours are similar enough are preserved as true positive links whilst the others are discarded as false positive links. The selector is an ad-hoc component that works as follows: it takes the filtered links and the supporting rules used to filter them; then, it performs a voting strategy regarding how many rules filtered the same link and another voting strategy regarding how many links were filtered by each rule; finally, a subset of the filtered links is selected and preserved as the rest are discarded.

Example 3.1 *Our running example from Section §B.2 presents two sample datasets that are based on the DBLP and the NSF datasets; described in Appendix §A.2. The resources are depicted in greyed boxes whose shapes encode their classes (i.e., the value of property* rdf:type*), the properties are represented as labelled arrows, and the literals are encoded as strings. The genetic component learns the following link rules in this scenario, which we represent using a Prolog-like notation for the sake of readability:*

$$\begin{aligned} r_1\text{: } \mathtt{link}(A,R) \text{ if } & \mathtt{rdf{:}type}(A) = \mathtt{dblp{:}Author}, \mathtt{rdf{:}type}(R) = \mathtt{nsf{:}Researcher}, \\ & N_A = \mathtt{dblp{:}name}(A), N_R = \mathtt{nsf{:}name}(R), \\ & S_1 = \mathtt{levenstein}(\mathtt{lfname}(N_A), \mathtt{lfname}(N_R)), \\ & \frac{S_1 - 0.80}{1.00 - 0.80} > 0. \end{aligned}$$

$$\begin{aligned} r_2\text{: } \mathtt{link}(A,P) \text{ if } & \mathtt{rdf{:}type}(A) = \mathtt{dblp{:}Article}, \mathtt{rdf{:}type}(P) = \mathtt{nsf{:}Paper}, \\ & T_A = \mathtt{dblp{:}title}(A), T_P = \mathtt{nsf{:}title}(P), \\ & \mathtt{jaccard}(\mathtt{lowercase}(T_A), \mathtt{lowercase}(T_P)), \\ & \frac{S_1 - 0.65}{1.00 - 0.65} > 0. \end{aligned}$$

where levenstein *and* jaccard *denote the well-known string similarity metrics (normalised to interval* $[0.00, 1.00]$*),* lfname *is a transformation metric that normalises people's names as "last name, first name", and* lowercase *is a transformation metric that changes a string into lowercase.*

Intuitively, link rule r_1 *is applied to a resource* A *of type* dblp:Author *and a resource* R *of type* nsf:Researcher*; it computes the normalised Levenshtein metric between the normalised names of the author and the researcher; if it is greater than* 0.80*, then the corresponding resources are linked. Link rule* r_2 *should now be easy to interpret: it is applied to a resource* A *of type* dblp:Article *and a resource* P *of type* nsf:Paper *and links them if the normalised Jaccard metric amongst the lowercase version of the title of article* A *and the title of paper* P *is greater than* 0.65*.*

Realise that link rule r_1 *links resources* dblp:weiwang *and* $\mathtt{nsf{:}weiwang}_1$ *or* dblp:binliu *and* nsf:binwliu*, which are true positive links, but also* dblp:weiwang *and* $\mathtt{nsf{:}weiwang}_2$*, which is a false positive link. In cases like this, the only way to make a difference between such resources is to analyse their neighbours, be them direct (e.g.,* dblp:weiwang *and* $\mathtt{dblp{:}article}_2$*) or transitive (e.g.,* $\mathtt{nsf{:}weiwang}_1$ *and* $\mathtt{nsf{:}paper}_2$*).*

3.2 Bootstrapping process

In this section we introduce our bootstrapping process, which consist of namely a filtering method, a similarity metric, and a method to select suitable

```
filterLinks(r, S, D1, D2, θ, μ, ρ) : K
  K := ∅
  (C1, C2) := (sourceClasses(r), targetClasses(r))
  L1 := apply(r, D1, D2)
  for r′ ∈ S do
    (C1′, C2′) := (sourceClasses(r′), targetClasses(r′))
    (P1, P2) := (findPath(C1, C1′, D1), findPath(C2, C2′, D2))
    L2 = apply(r′, D1, D2)
    for (p1, p2) ∈ P2 × P2 do
      for (a, b) ∈ L1 do
        (A, B) := (findResources(a, p1, D1), findResources(b, p2, D2))
        E := L2 ∩ (A × B)
        w := computeSimilarity(A, B, E)
        if w ≥ θ then
          K := K ∪ {(a, b)}
        end
      end
    end
  end
  K := selectLinks(K, μ, ρ)
end
```

Algorithm 3.1: *Method to filter links.*

links between resources generated as result of this process. In following subsections we aim at introducing each of these elements that conform our bootstrapping proposal

3.2.1 Filtering links

Algorithm §3.1 presents the method to filter links. It works on a link rule r, a set of supporting link rules S, a source dataset D_1, a companion dataset D_2, and a threshold θ that we explain later. It returns K, which is the subset of links produced by base link rule r that seem to be true positive links.

The method first initialises K to an empty set, stores the source and the target classes of the base link rule in sets C_1 and C_2, respectively, and the links that result from applying it to the source and the companion datasets in set L_1.

The main loop then iterates over the set of supporting link rules using variable r'. In each iteration, it first computes the sets of source and target

classes involved in link rule r', which are stored in variables C_1' and C_2', respectively; next, it finds the set of paths P_1 that connect the source classes in C_1 with the source classes in C_1' in dataset D_1; similarly, it finds the set of paths P_2 that connect the target classes in C_2 with the target classes in C_2' in dataset D_2. By path between two sets of classes, we mean a sequence of object properties that connect resources with the first set of classes to resources with the second set of classes, irrespective of their direction. Simply put: the idea is to find the way to connect the resources linked by the base link rule with the resources linked by the supporting link rule, which is done by the intermediate and the inner loops.

The intermediate loop iterates over the set of pairs of paths (p_1, p_2) from the Cartesian product of P_1 and P_2. If there is at least a pair of such paths, it then means that the resources involved in the links returned by base link rule r might have some neighbours that might be linked by supporting link rule r'.

The inner loop iterates over the collection of links (a, b) in set L_1. It first finds the set of resources A that are reachable from resource a using path p_1 in source dataset D_1 and the set of resources B that are reachable from resource b using path p_2 in the companion dataset D_2. Next, the method applies supporting link rule r' to the source and the companion dataset and intersects the resulting links with $A \times B$ so as to keep resources that are not reachable from a or b apart; the result is stored in set E. It then computes the similarity of sets A and B; intuitively, the higher the similarity, the more likely that resources a and b refer to the same real-world entity. If the similarity is equal or greater than threshold θ, then link (a, b) is added to set K; otherwise, it is filtered out. When the main loop finishes, set K contains the collection of links that involve neighbours that are similar enough according to the supporting rules.

We do not provide any additional details regarding the algorithms to find paths or resources since they can be implemented using Dijkstra's algorithm to find the shortest paths in a graph. Computing the similarity coefficient is a bit more involved, so we devote a subsection to this ancillary method below.

Example 3.2 *Relying on our running example presented in Section §B.2, and assuming the link rule* r_1 *as the base link rule, i.e., we are interested in linking authors and researchers, and we use link rule* r_2 *as the support link rule, i.e., we consider both their articles and papers. Their source classes are* $C_1 = \{\texttt{dblp:Author}\}$ *and* $C_1' = \{\texttt{dblp:Article}\}$*, and their target classes are* $C_2 = \{\texttt{nsf:Researcher}\}$ *and* $C_2' = \{\texttt{nsf:Paper}\}$*. Link rule* r_1 *returns the following set of links* L_1*:*

$(\texttt{dblp:}weiwang, \texttt{nsf:}weiwang_1)$,

$(\text{dblp:}weiwang, \text{nsf:}weiwang_2)$, *and*
$(\text{dblp:}binliu, \text{nsf:}binwliu)$.

Note that the first and the third links are true positive links, but the second one is a false positive link. Link rule r_2 *returns the following set of links* L_2*, which are true positive links:*

$(\text{dblp:}article_1, \text{nsf:}paper_3)$,
$(\text{dblp:}article_2, \text{nsf:}paper_2)$,
$(\text{dblp:}article_4, \text{nsf:}paper_2)$, *and*
$(\text{dblp:}article_5, \text{nsf:}paper_5)$.

The sets of paths amongst the source and target classes of r_1 *and* r_2 *are, respectively,* $P_1 = \{\langle \text{dblp:}writtenBy \rangle\}$ *and* $P_2 = \{\langle \text{nsf:}leads, \text{nsf:}supports \rangle\}$*. Now, the links in* L_1 *are scanned and the resources that can be reached from the resources involved in each link using the previous paths are fetched.*

Link $l_1 = (\text{dblp:}weiwang, \text{nsf:}weiwang_1)$ *is the first to be analysed. The method finds* $A = \{\text{dblp:}article_1, \text{dblp:}article_2, \text{dblp:}article_3, \text{dblp:}article_4\}$ *by following resource* dblp:*weiwang through path* $\langle \text{dblp:}writtenBy \rangle$*; it also finds* $B = \{\text{nsf:}paper_1, \text{nsf:}paper_2, \text{nsf:}paper_3\}$ *by following resource* nsf:$weiwang_1$ *through path* $\langle \text{nsf:}leads, \text{nsf:}supports \rangle$*. Now, supporting link rule* r_2 *is applied and the results are intersected with* $A \times B$ *so as to keep links that are related to* l_1 *only; the result is the following set of links* E*:*

$(\text{dblp:}article_1, \text{nsf:}paper_3)$,
$(\text{dblp:}article_2, \text{nsf:}paper_2)$, *and*
$(\text{dblp:}article_4, \text{nsf:}paper_2)$.

Then, the similarity of A *and* B *in the context of* E *is computed, which returns* 0.67*; intuitively, there are chances that* l_1 *is a true positive link.*

Link $l_2 = (\text{dblp:}weiwang, \text{nsf:}weiwang_2)$ *is the next to be analysed. The method finds* $A = \{\text{dblp:}article_1, \text{dblp:}article_2, \text{dblp:}article_3, \text{dblp:}article_4\}$ *by following resource* dblp:*weiwang through path* $\langle \text{dblp:}writtenBy \rangle$*; next, it finds* $B = \{\text{nsf:}paper_4\}$ *by following resource* nsf:$weiwang_2$ *through path* $\langle \text{nsf:}leads, \text{nsf:}supports \rangle$*. Now, supporting link rule* r_2 *is applied and the result is intersected with* $A \times B$*, which results in* $E = \emptyset$*. In such a case, the similarity is zero, which intuitively indicates that it is very likely that* l_2 *is a false positive link.*

Link $l_3 = (\text{dblp:}binliu, \text{nsf:}binwliu)$ *is analysed next. The method finds* $A = \{\text{dblp:}article_5\}$ *by following resource* dblp:*binliu through path* $\langle$dblp:

```
computeSimilarity(A, B, E) : d
  A′ := reduce(A, E)
  B′ := reduce(B, E)
  W := intersect(A′, B′, E)
  d := |W|/ min{|A′|, |B′|})
end
```

Algorithm 3.2: *Method to compute similarity.*

writtenBy⟩*; next, it finds* $B = \{\text{nsf:paper}_5\}$ *by following resource* nsf:binwliu *through path* ⟨nsf:leads, nsf:supports⟩*. Now, supporting link rule* r_2 *is applied and the result is intersected with* $A \times B$*, which results in* $E = \{(\text{dblp:article}_5, \text{nsf:paper}_5)\}$*. The similarity is now* 1.00*, i.e., it is very likely that link* l_3 *is a true positive link.*

Assuming that $\theta = 0.50$*, for instance, the* filterLinks *method would then return set* $K = \{(\text{dblp:}weiwang, \text{nsf:}weiwang_1), (\text{dblp:binliu}, \text{nsf:binwliu})\}$*. Note that the previous value of* θ *is intended for illustration purposes only because the running example must necessarily have very little data.*

3.2.2 Computing neighbours similarity

Algorithm §3.2 shows our method to compute similarities. Its input consists of sets A and B, which are two sets of resources, and E, which is a set of links between them. It returns the Szymkiewicz-Simpson overlapping coefficient, namely:

$$\text{overlap}(A, B) = \frac{|A \cap B|}{\min\{|A|, |B|\}}$$

The previous formula assumes that there is an implicit equality relation to compute $A \cap B$, $|A|$, or $|B|$. In our context, this relation must be inferred from the set of links E by means of Warshall's algorithm to compute the reflexive, commutative, transitive closure of relation E, which we denote as $E^\star$.

The method to compute similarities relies on two ancillary functions, namely: reduce, which given a set of resources X and a set of links E returns a set whose elements are subsets of X that are equal according to $E^\star$, and intersect, which given two reduced sets of resources X and Y and a set of

links E returns the intersection of X and Y according to $E^\star$. Their definitions are as follows:

$$\begin{aligned} \mathtt{reduce}(X, E) &= \{W \mid W \propto W \subseteq X \wedge W \times W \subseteq E^\star)\} \\ \mathtt{intersect}(X, Y, E) &= \{W \mid W \propto W \subseteq X \wedge \exists W' : W' \subseteq Y \wedge W \times W' \in E^\star\} \end{aligned}$$

where $X \propto \phi$ denotes the maximal set X that fulfils predicate ϕ, that is:

$$X \propto \phi \Leftrightarrow \phi(X) \wedge (\nexists X' : X \subseteq X' \wedge \phi(X'))$$

The method to compute similarities then works as follows: it first reduces the input sets of resources A and B according to the set of links E; it then computes the intersection of both reduced sets; finally, it computes the similarity using Szymkiewicz-Simpson's formula on the reduced sets.

Example 3.3 *Analysing link* $l_1 = (\mathtt{dblp{:}weiwang}, \mathtt{nsf{:}weiwang}_1)$ *results in sets* $A = \{\mathtt{dblp{:}article}_1, \mathtt{dblp{:}article}_2, \mathtt{dblp{:}article}_3, \mathtt{dblp{:}article}_4\}$, $B = \{\mathtt{nsf{:}paper}_1, \mathtt{nsf{:}paper}_2, \mathtt{nsf{:}paper}_3\}$, *and* $E = \{(\mathtt{dblp{:}article}_1, \mathtt{nsf{:}paper}_3), (\mathtt{dblp{:}article}_2, \mathtt{nsf{:}paper}_2), (\mathtt{dblp{:}article}_4, \mathtt{nsf{:}paper}_2)\}$. *If* E *is interpreted as an equality relation, then it is straightforward to realise that* $\mathtt{dblp{:}article}_2$ *and* $\mathtt{dblp{:}article}_4$ *can be considered equal, because* $\mathtt{dblp{:}article}_2$ *is equal to* $\mathtt{nsf{:}paper}_2$ *and* $\mathtt{nsf{:}paper}_2$ *is equal to* $\mathtt{dblp{:}article}_4$. *Thus, set* A *is reduced to* $A' = \{\{\mathtt{dblp{:}article}_1\}, \{\mathtt{dblp{:}article}_2, \mathtt{dblp{:}article}_4\}, \{\mathtt{dblp{:}article}_3\}\}$ *and set* B *is reduced to* $B' = \{\{\mathtt{nsf{:}paper}_1\}, \{\mathtt{nsf{:}paper}_2\}, \{\mathtt{nsf{:}paper}_3\}\}$. *As a conclusion,* $|A' \cap B'| = |\{\{\mathtt{dblp{:}article}_1, \mathtt{nsf{:}paper}_3\}, \{\mathtt{dblp{:}article}_2, \mathtt{dblp{:}article}_4, \mathtt{nsf{:}paper}_2\}\}| = 2$, $|A'| = 3$, *and* $|B'| = 3$*; so the similarity is* 0.67.

When link $l_2 = (\mathtt{dblp{:}weiwang}, \mathtt{nsf{:}weiwang}_2)$ *is analysed,* $A = \{\mathtt{dblp{:}article}_1, \mathtt{dblp{:}article}_2, \mathtt{dblp{:}article}_3, \mathtt{dblp{:}article}_4\}$, $B = \{\mathtt{nsf{:}paper}_4\}$, *and* $E = \emptyset$. *Since the equality relation* $E^\star$ *is then empty, the similarity is zero because the intersection between the reductions of sets* A *and* B *is empty.*

Regarding link $l_3 = (\mathtt{dblp{:}binliu}, \mathtt{nsf{:}binwliu})$, *the method first computes* $A = \{\mathtt{dblp{:}article}_5\}$, *then* $B = \{\mathtt{nsf{:}paper}_5\}$, *and, finally,* $E = \{(\mathtt{dblp{:}article}_5, \mathtt{nsf{:}paper}_5)\}$. *As a conclusion,* $|A' \cap B'| = |\{\{\mathtt{dblp{:}article}_5, \mathtt{nsf{:}paper}_5\}\}| = 1$, $|A'| = 1$, *and* $|B'| = 1$, *where* A' *and* B' *denote, respectively, the reductions of sets* A *and* B*; so the similarity is* 1.00.

3.2.3 Selecting links

Algorithm §3.3 shows the method to select the best links out of a set of correspondences. Its input consists of a set of correspondences K, a threshold μ to the minimum number of times that a link must have been selected by

$selectLinks(K, \mu, \rho) : L$
$M := \{(s,t) \mid \exists r' : (s,t,r') \in K\}$
$S := \{r' \mid \exists s,t : (s,t,r') \in K\}$
$g := \mu \max\{n \mid \exists a,b : (a,b) \in M \wedge n = |rules(a,b,K)|\}$
$d := \rho \max\{m \mid \exists r' : r' \in S \wedge m = |links(r',K)|\}$
$U := \{(a,b) \mid (a,b) \in M \wedge |rules(a,b,K)| \geq g\}$
$V := \{r' \mid r' \in S \wedge links(r',K) \geq d\}$
$L := \{(a,b) \mid (a,b) \in M \wedge ((a,b) \in U \wedge rules(a,b,K) \cap V \neq \emptyset)\}$
end

Algorithm 3.3: *Method to select filtered links.*

a supporting link rule so that it can be selected by this method (top links), and an additional threshold ρ to the minimum number of times that a supporting link rule must have selected a link so that links that have been selected by that link rule can be selected by this method (top link rules) even if they are not top links.

This method relies on two ancillary functions, namely: $links$, which given a supporting link rule r' returns the set of links that it has selected, and $rules$, which given a link (a,b) returns the set of link rules that have selected it. The previous functions are formally defined as follows:

$$links(r',K) = \{(a,b) \mid \exists r' : (a,b,r') \in K\}$$
$$rules(a,b,K) = \{r' \mid \exists a,b : (a,b,r') \in K\}$$

The method to select links first projects the set of correspondences K onto the set of links M and the set of supporting link rules S. It then computes g as a percentage, according to μ, of the maximum number of link rules that have selected each candidate link; it also computes d as a percentage, according to ρ, of the maximum number of links that a support link rule has selected as candidates. Next, it computes the set of top links U as the set of links in M that have been selected by at least g link rules; similarly, it computes the set of top link rules V as the set of link rules in S that have selected at least d links. The resulting set of links L is computed as the subset of links in M that are either top links or have been selected by top link rules.

3.3 Experimental analysis

In this section, we first describe our experimental environment and then comment on our results obtained in the experiments. The goal of our experi-

mentation is to show that Teide significantly improves the precision of a link rule in the different scenarios studied without a significant drop in the recall.

3.3.1 Experimental environment

Implementation. We implemented our proposal[†1] with Java 1.8 and the following components: Jena TDB 3.2.0 to work with RDF data, ARQ 3.2.0 to work with SPARQL queries, and Simmetrics 1.6.2, SecondString 2013-05-02, and JavaStringSimilarity 1.0.1 to compute string similarities.

Running experiments. We run our experiments in the facility described in Appendix §A.1. In addition, we used Genlink as baseline to learn link rules, which was implemented relying on our framework presented in the previous chapter. We chose this proposal since it was the one that achieved the worst positions in the statistical ranking that we performed and we aim at showing that Teide improves even the last-ranked proposal. However, any proposal from the state of the literature could be used.

Measures. On the one hand, we explored a large portion of the parameter space to establish optimal values for θ, μ, ρ in each scenario. On the other hand, we measured the number of links returned by each proposal ($Links$), precision (P), recall (R), and the F_1 score (F_1). We also computed the normalised differences in precision (ΔP), recall (ΔR), and F_1 score (ΔF_1), which measure the ratio from the difference found between the baseline and our proposal and the maximum possible difference for each performance measure.

Datasets sizes. Our experiments rely on some of the scenarios described in Appendix §A.2. For these experiments, we used the following datasets:

- In scenario Restaurants, we used 113 and 752 resources in each sub-dataset, respectively. In addition, we provided 112 $owl{:}sameAs$ links and 84 864 $owl{:}differentFrom$ links.
- In scenario Persons1, we used 500 resources per sub-dataset; from which we provided 500 $owl{:}sameAs$ links and $249,500$ $owl{:}differentFrom$ links.

[†1]The prototype is available at https://github.com/AndreaCimminoArriaga/TeidePlus.

- In scenario Persons2, the sub-dataset have 400 and 600 resources, respectively. In addition, we provided 400 owl:sameAs links and 239,600 owl:differentFrom links.
- In scenario Publications, the sub-dataset consisted of 108 resources from the RAE and 98 resources from the Newcastle. In addition, we provided 108 owl:sameAs links and 10,476 owl:differentFrom links.
- In scenario Authors, the sub-dataset consisted of 9,076 resources from the DBLP, respectively; we provided 9,076 owl:sameAs links and 82,364 , 700 owl:differentFrom links.
- In scenario Researchers, the sub-dataset consisted of 100 resources from the DBLP and 130 resources from the NSF. In addition, we provided 33 owl:sameAs links and 12,967 owl:differentFrom links.
- In scenario Films, the sub-dataset consisted of 691 resources from the BBC and 445 resources from the DBpedia. In addition, we provided 445 owl:sameAs links and 307,050 owl:differentFrom links.
- In scenario Movies, the sub-dataset consisted of 96 resources from the DBpedia and 101 resources from the IMDb. In addition, we provided 58 owl:sameAs links and 9,638 owl:differentFrom links.
- In scenario Doremus16-9ht, the sub-dataset have 40 and 40 resources, respectively. In addition, we provided 32 owl:sameAs links and 1,584 owl:differentFrom links.
- In scenario Doremus16-fp, the sub-dataset have 85 and 41 resources, respectively. In addition, we provided 41 owl:sameAs links and 3,444 owl:differentFrom links.
- In scenario Doremus17-ht, the sub-dataset have 238 and 238 resources, respectively. In addition, we provided 47 owl:sameAs links and 56,597 owl:differentFrom links.
- In scenario Doremus17-fp, the sub-dataset have 75 and 75 resources, respectively. In addition, we provided 15 owl:sameAs links and 5,610 owl:differentFrom links.

3.3.2 Experimental results

The results are presented in Table §3.1. We analyse them in terms of precision, recall, and the F_1 score.

	Genlink							Teide							
Scenario	links	P	R	F_1	θ	μ	ρ	links	P	R	F_1	ΔLinks	ΔP	ΔR	ΔF_1
Researchers	127	0.25	0.97	0.40	0.01	0.02	0.74	33	0.97	0.97	0.97	-94	0.96	0.00	0.95
Authors	78,348	0.12	1.00	0.21	1.00	0.07	0.03	9,069	1.00	1.00	1.00	-69,279	1.00	0.00	1.00
Films	525	0.85	1.00	0.92	0.10	0.00	0.03	461	0.96	1.00	0.98	-64	0.74	0.00	0.72
Movies	118	0.27	0.55	0.36	0.60	0.00	0.03	41	0.68	0.48	0.57	-77	0.57	-0.12	0.32
Publications	404	0.22	0.82	0.35	0.30	0.13	0.03	68	0.72	0.45	0.56	-336	0.64	-0.45	0.32
Restaurants	103	0.90	0.83	0.87	0.10	0.08	0.58	96	0.97	0.83	0.89	-7	0.69	0.00	0.19
Persons1	1655	0.29	0.96	0.45	0.01	0.01	0.01	691	0.69	0.96	0.81	-964	0.57	0.00	0.65
Persons2	1340	0.22	0.74	0.34	0.01	0.01	0.01	451	0.65	0.74	0.69	-889	0.55	-0.01	0.53
Doremus16-9ht	29	0.83	0.75	0.79	0.01	0.01	0.01	24	1.00	0.75	0.86	-5	1.00	0.00	0.33
Doremus16-fp	221	0.14	0.76	0.24	0.01	0.01	0.01	44	0.61	0.66	0.64	-177	0.55	-0.13	0.52
Doremus17-ht	1880	0.03	1.00	0.05	0.01	0.01	0.01	127	0.34	0.91	0.49	-1753	0.32	-0.09	0.47
Doremus17-fp	808	0.02	1.00	0.04	0.01	0.01	0.01	9	0.67	0.40	0.50	-799	0.66	-0.60	0.48
												Average Δ	0.69	-0.12	0.54
												Iman-Davenport's test	0.01	0.19	0.01

Table 3.1: *Experimental effectiveness and efficiency.*

It is clear that our technique improves the precision of the rules learnt by GenLink in every scenario. In average, the difference in precision is 69%. The worst improvement is 32% in the Doremus17-ht scenario since these datasets are clearly unbalanced: resources in one sub-dataset have such a different representation from the resources in the other that link rules are not able to capture their similarity; this obviously makes it impossible for our proposal to find enough context to make a decision. The best improvement is 100% in the Researchers scenario since there are $9,069$ authors with very similar names, which makes it almost impossible for GenLink to generate rules with good precision building solely on the attributes of the resources.

The normalised difference of recall ΔR shows that our proposal generally retains the recall of the link rules learnt by GenLink, except in the Movies, Publications, Doremus16-fp, and Doremus17-fp scenarios. The problem with the previous scenarios was that there are many incomplete resources, that is, many resources without neighbours. For instance, there are 43 papers in the Newcastle dataset that are not related to any authors. The incompleteness of data has also a negative impact on the recall of the base link rules. In our prototype, we are planning on implementing a simple check to identify incomplete resources so that the links in which they are involved are not discarded as false positives, but identified as cases on which our proposal cannot make a sensible decision.

We also studied ΔF_1, which denotes the normalised difference in F_1 score. Note that it is 54% in average, which is a large difference. However, without a statistical analysis we cannot conclude that Teide improves the precision of the rules without a significant drop in the recall.

3.3.3 Statistical analysis

The statistical analysis is reported under the table displayed in Table §3.1, we applied the Iman-Iman-Davenport's test with an alpha of 0.05 to check if there are significant differences between the precision, entailing that our proposal improves the precision, and there are no significant differences in the recall, entailing that our proposal may have a bit of less recall but without a relevant significance.

The p-value computed by Iman-Davenport's test in terms of precision is 0.01; since it is clearly smaller than the standard confidence level, we can interpret it as a strong indication that there is enough evidence in our experimental data to confirm the hypothesis that our proposal works better than the baseline regarding precision.

Iman-Davenport's test for recall returns 0.19 as the corresponding p-value; since it is larger than the standard confidence level, it may be interpreted as a strong indication that the differences in recall found in our experiments are not statistically significant. In other words, the cases in which data are that incomplete do not seem to be common-enough for them to have an overall impact on our proposal.

The corresponding Iman-Davenport's p-value for the F_1 is 0.01, which can be interpreted as a strong indication that the difference is significant from a statistical point of view. Overall, this result confirms that our proposal helps improve precision without degrading recall and that the improvement in precision is enough for the F_1 score to improve significantly.

At the light of these results, we can conclude that Teide improves significantly the precision of the link rules, without a significant difference in the recall.

3.4 Conclusions

Data inter-operability of business systems based on Web of Data requires to link the resources that are available in different datasets and represent the same real-world entities. Such links are generated by link rules that take the values of the attributes of the resources into account, but not their neighbours, which sometimes results in false positives that have a negative impact on their precision. We have presented a novel proposal called Teide that relies on a genetic programming proposal to learn a set of link rules and then boosts them, which has proven to improve the overall F_1 score.

Chapter 4

Sorbas: Learning Context-Aware Link Rules

This chapter introduces our proposal to learn link rules that consider the context of data, which has been proved that increases the precision of the rules without a significant drop in their recall. It is organised as follows: Section §4.1 introduces the context of our proposal; Section §4.2 describes and explains our proposal; Section §4.3 explains and reports the experiments conducted with our implementations; and finally, Section §4.4 recaps on the conclusions drawn from our experiments.

4.1 Introduction

The previous chapter introduced our proposal Teide to bootstrap link rules, and take advantage of the context of the resources been linked. However, we realised that since Teide does not export any rule, it just applies them, there is no re-usability and, in addition, the execution time was high due to the fact that not all the supporting link rules are useful to link resources, and those that are not, are applied constantly. As a result, a lot of comparisons that are not necessary are performed by Teide, which increases its execution time.

In this chapter, we present an approach known as Sorbas that learns context-aware link rules building on the acontextual rules learnt by any proposal from the literature, or just provided by a user. By context-aware, we mean that the rule takes into account the data properties of the resources being linked and the data properties of their neighbours. By learning the rules, a user may rely on a training set in which learning a context-aware link rule is computationally affordable, and once the rule is learnt, use our approach Teide or Sorbas to apply the learnt rule. As a result, the applying more precise rules will take less time.

4.2 Learning process

The input to our learning method is a base rule R, a set of support rules S, and two datasets D_1 and D_2. We assume that the rules have been learnt with the first component, that is, they are acontextual; we also assume that the datasets provide resources that are representative enough of the resources that we wish to link. Our goal is to learn a context-aware rule that combines base rule R with a subset of support rules S to improve the precision when linking similar datasets. Our proposal works in three steps: it first learns a set of correspondences, next filters some of them out, and then instantiates a template to produce the resulting rule.

The first step learns a set of correspondences K, which are tuples of the form (A, B, T, P_1, P_2). In the previous tuple, A and B denote two resources that are linked by means of base rule R; P_1 and P_2 denote two paths, that is, two sequences of object properties that relate resources A and B to two subsets of direct or indirect neighbours; and T denotes a support rule that establishes some links amongst the previous subsets of neighbours. There can be

many correspondences, but the method filters out the ones in which the subsets of direct or indirect neighbours cannot be considered similar enough according to the links found by support rule T. Intuitively, the correspondences keep the links whose context can be considered similar enough not to be false positive links.

The second step consists in selecting the support rules in the set of correspondences K that have generated enough links. We use a threshold γ that is computed using grid parameter search to set the minimum number of links that a support rule must have generated so that it can be selected. We first compute a set with the counters of links in K that have been generated by each support rule T given two paths P_1 and P_2, that is, we compute:

$$P = \{C \mid \exists T, P_1, P_2 : C = |links(K, T, P_1, P_2)|\}$$

where $links$ is defined as follows:

$$links(K, T, P_1, P_2) = \{(A, B) \mid (A, B, T, P_1, P_2) \in K\}$$

The set of support rules selected is then defined as the set of support rules that generate at least γ percent the maximum number of links generated by a support rule, that is:

$$V = \{(T, P_1, P_2) \mid |links(K, T, P_1, P_2)| \geq \gamma \max P\}$$

Note that V stores triplets in which each support rule is accompanied by two paths; the reason is that a link rule may generate many links, but we are interested in links amongst resources that are directly or indirectly related to the original resources whose linkage must be decided. (In a previous instance-driven approach to find links, we realised that this is a good heuristic [17].)

The final step consist in generating the resulting context-aware rule, which is an instantiation of the following template:

$$\begin{aligned} link(A, B) \text{ if } & \mathbf{R}(A, B) \wedge \exists (T, P_1, P_2) \in \mathbf{V} : \\ & X = findResources(A, P_1) \wedge Y = findResources(B, P_2) \wedge \\ & E = \{(U, V) \mid U \in X \wedge V \in Y \Rightarrow T(U, V)\} \wedge \\ & computeSimilarity(X, Y, E) \geq \theta \end{aligned}$$

The previous template is a general model for which our procedure learns the following parameters: $\mathbf{R}$, which denotes a base rule, $\mathbf{V}$, which denotes the set of support rules and paths selected previously, and θ, which is a similarity threshold that we learn by means of grid parameter search.

```
computeCorrespondences(R, S, D1, D2, θ) : K
  K := ∅
  (RS, RT) := (sourceClasses(R), targetClasses(R))
  L1 := apply(R, D1, D2)
  for T ∈ S do
    (TS, TT) := (sourceClasses(T), targetClasses(T))
    (Q1, Q2) := (findPaths(RS, TS, D1), findPaths(RT, TT, D2))
    L2 = apply(T, D1, D2)
    for (P1, P2) ∈ Q1 × Q2 do
      for (A, B) ∈ L1 do
        (X, Y) := (findResources(A, P1, D1), findResources(B, P2, D2))
        E := L2 ∩ (X × Y)
        if computeSimilarity(X, Y, E) ≥ θ then
          K := K ∪ {(A, B, T, P1, P2)}
        end
      end
    end
  end
end
```

Algorithm 4.1: *Method to learn correspondences.*

Intuitively, resources A and B can be linked if they are linked by the base rule and their neighbourhoods are similar enough. The neighbours are similar enough if there is at least a support rule T with paths P_1 and P_2, such that: let X denote the neighbours of resource A by following path P_1, let Y denote the neighbours of resource B by following path P_2, and let E be the set of links that support rule T finds amongst X and Y; the neighbours are similar enough if X and Y are deemed similar enough according to the links that the supporting rule has found.

Method `findResources` is very simple, so we do not provide any additional details. In the subsections below, we delve into the intricacies of computing correspondences and similarities.

4.2.1 Computing correspondences

Figure §4.1 provides the pseudo-code to the method to learn correspondences. It works on a base rule R, a set of support rules S, two datasets D_1 and D_2, and a similarity threshold θ. It returns a set of correspondences K.

The method first initialises K to an empty set, stores the source and the target classes of the base rule in sets R_S and R_T, respectively, and the links that result from applying it to the input datasets in set L_1.

The main loop then iterates over the set of support rules using variable T. In each iteration, it first computes the sets of source and target classes involved in rule T, which are stored in variables R_S and R_T, respectively; next, it finds the set of paths Q_1 that connect the source classes in R_S with the source classes in T_S in dataset D_1; similarly, it finds the set of paths Q_2 that connect the target classes in R_T with the target classes in T_T in dataset D_2. Simply put: the idea is to find the paths to relate the resources linked by the base rule with the resources linked by the support rule, which is done by the intermediate and the inner loops.

The intermediate loop iterates over the set of pairs of paths (P_1, P_2) from the Cartesian product of Q_1 and Q_2. If there is at least a pair of such paths, it then means that the resources involved in the links returned by base rule R might have some neighbours that might be linked by support rule T. The inner loop iterates over the collection of links (A, B) in set L_1, that is, the links returned by the base rule. It first finds the set of resources X that are related to resource A using path P_1 in dataset D_1 and the set of resources Y that are related to resource B using path P_2 in dataset D_2. Next, the method applies support rule T to datasets D_1 and D_2 and intersects the resulting links with $X \times Y$ in order to filter out the resources that are not neighbours of A or B; the result is stored in set E. It then computes the similarity of sets A and B. If it is greater than or equal to threshold θ, then correspondence (A, B, T, P_1, P_2) is added to set K; otherwise, it is filtered out.

Method `findPaths` basically resorts to Dijkstra's well-known algorithm to find the shortest paths in a graph. We do not provide any additional details regarding methods `sourceClasses`, `targetClasses`, `findResources`, and `apply` because they are straightforward.

4.2.2 Computing similarity

Figure §4.2 shows our method to compute similarity. Its input consists of sets X and Y, which are two sets of resources, and E, which is a set of links between them. It returns the Szymkiewicz-Simpson overlapping coefficient, namely:

$$\mathtt{overlap}(X, Y) = \frac{|X \cap Y|}{\min\{|X|, |Y|\}}$$

```
computeSimilarity(X, Y, E) : D
  X′ := reduce(X, E)
  Y′ := reduce(Y, E)
  W := intersect(X′, Y′, E)
  D := |W|/ min{|X′|, |Y′|})
end
```

Algorithm 4.2: *Method to compute similarity.*

The previous formula assumes that there is an implicit equality relation to compute $X \cap Y$, $|X|$, or $|Y|$. In our context, this relation must be inferred from the set of links E by means of Warshall's algorithm to compute the reflexive, commutative, transitive closure of relation E, which we denote as $E^\star$.

The method to compute similarities relies on two ancillary methods, namely: $reduce$, which given a set of resources X and a set of links E returns a set whose elements are subsets of X that are equal according to $E^\star$, and $intersect$, which given two reduced sets of resources X and Y and a set of links E returns the intersection of X and Y according to $E^\star$. Their definitions are as follows:

$$reduce(X, E) = \{W \mid W \propto W \subseteq X \wedge W \times W \subseteq E^\star\}$$
$$intersect(X, Y, E) = \{W \mid W \propto W \subseteq X \wedge \exists W' : W' \subseteq Y \wedge W \times W' \in E^\star\}$$

where $X \propto \phi$ denotes the maximal set X that fulfils predicate ϕ, that is:

$$X \propto \phi \Leftrightarrow \phi(X) \wedge (\nexists X' : X \subseteq X' \wedge \phi(X'))$$

The method to compute similarities then works as follows: it first reduces the input sets of resources X and X according to the set of links E; it then computes the intersection of both reduced sets; finally, it computes the similarity using Szymkiewicz-Simpson's formula on the reduced sets.

4.2.3 Illustration

The running example introduces in Section §B.2 presents an excerpt of the DBLP dataset and an excerpt of the NSF dataset. The first component learns the following rules in this scenario, where $levenshtein$ and $jaccard$ are the well-known string similarity functions, $lfname$ is a function that transforms names into format "last name, first name", and $lowercase$ is a function that changes a string into lowercase:

r_1: $link(A, R)$ if $rdf{:}type(A) = dblp{:}Author, rdf{:}type(R) = nsf{:}Researcher,$
$N_A = dblp{:}name(A), N_R = nsf{:}name(R),$
$S_1 = levenstein(lfname(N_A), lfname(N_R)),$
$\frac{S_1 - 0.87}{1.00 - 0.87} \geq 0.$

r_2: $link(A, P)$ if $rdf{:}type(A) = dblp{:}Article, rdf{:}type(P) = nsf{:}Paper,$
$T_A = dblp{:}title(A), T_P = nsf{:}title(P),$
$S_1 = jaccard(lowercase(T_A), lowercase(T_P)),.$
$\frac{S_1 - 0.65}{1.00 - 0.65} \geq 0.$

r_3: $link(A, U)$ if $rdf{:}type(A) = dblp{:}Affiliation, rdf{:}type(U) = nsf{:}Award,$
$T_A = dblp{:}name(A), T_U = nsf{:}uid(U),$
$S_1 = jaccard(lowercase(T_A), lowercase(T_U)),$
$\frac{S_1 - 0.95}{1.00 - 0.95} \geq 0.$

Intuitively, rule r_1 is applied to a resource A of type $dblp{:}Author$ and a resource R of type $nsf{:}Researcher$; it computes the Levenshtein similarity between the names of the author and the researcher after transforming them; if it is at least 0.80, then the corresponding resources are linked. Rules r_2 and r_3 should now be easy to understand.

Let us analyse the case in which the base rule is r_1 and the support rules are r_2 and r_3; that is, we are interested in linking DBLP authors and NSF researchers. Rule r_1 returns the following links: {($dblp{:}weiwang$, $nsf{:}weiwang_1$), ($dblp{:}weiwang$, $nsf{:}weiwang_2$), ($dblp{:}binliu$, $nsf{:}binwliu$)}; note that the first and the third links are true positive links, but the second one is a false positive link. Rule r_2 returns the following links: {($dblp{:}article_1$, $nsf{:}paper_3$), ($dblp{:}article_2$, $nsf{:}paper_2$), ($dblp{:}article_4$, $nsf{:}paper_2$), ($dblp{:}article_5$, $nsf{:}paper_5$) }, which are true positive links. Finally rule r_3 returns {($dblp{:}affiliation_1$, $nsf{:}award_4$)}, which are also true positive links. The sets of paths between the source and target classes of r_1 and r_2 are $\{\langle dblp{:}writtenBy\rangle\}$ and $\{\langle nsf{:}leads, nsf{:}supports\rangle\}$. Furthermore, the paths between the source and target classes of r_1 and r_3 are $\{\langle dblp{:}relatedTo\rangle\}$ and $\{\langle nsf{:}leads\rangle\}$, respectively.

Link $l_1 = (dblp{:}weiwang, nsf{:}weiwang_1)$ is analysed first. The method finds $X = \{dblp{:}article_1, dblp{:}article_2, dblp{:}article_3, dblp{:}article_4\}$ by following resource $dblp{:}weiwang$ through path $\langle dblp{:}writtenBy\rangle$; similarly, it finds $Y = \{nsf{:}paper_1, nsf{:}paper_2, nsf{:}paper_3\}$ by following resource $nsf{:}weiwang_1$ through path $\langle nsf{:}leads, nsf{:}supports\rangle$. Now, support rule r_2 is

applied and the results are intersected with $X \times Y$ so as to keep links that are related to l_1 only; the result is $E = \{(\text{dblp:article}_1, \text{nsf:paper}_3), (\text{dblp:article}_2, \text{nsf:paper}_2), (\text{dblp:article}_4, \text{nsf:paper}_2)\}$. Then, the similarity of X and Y in the context of E is computed, which returns 0.67; intuitively, there are chances that l_1 is a true positive link.

Link $l_2 = (\text{dblp:}weiwang, \text{nsf:}weiwang_2)$ is analysed next. The method finds $X = \{\text{dblp:article}_1, \text{dblp:article}_2, \text{dblp:article}_3, \text{dblp:article}_4\}$ by following resource dblp:*weiwang* through path $\langle \text{dblp:}writtenBy \rangle$; next, it finds $Y = \{\text{nsf:paper}_4\}$ by following resource $\text{nsf:}weiwang_2$ through path $\langle \text{nsf:}leads, \text{nsf:supports} \rangle$. Now, support rule r_2 is applied and the result is intersected with $X \times Y$, which results in $E = \emptyset$. In this case, the similarity is zero, which intuitively indicates that it is very likely that l_2 is a false positive link.

Link $l_3 = (\text{dblp:binliu}, \text{nsf:}binwliu)$ is analysed next. The method finds $A_1 = \{\text{dblp:article}_5\}$ by following resource dblp:binliu through path $\langle \text{dblp:}writtenBy \rangle$; next, it finds $Y_1 = \{\text{nsf:paper}_5\}$ by following resource nsf:*binwliu* through path $\langle \text{nsf:}leads, \text{nsf:supports} \rangle$. Now, support rule r_2 is applied and the result is intersected with $X_1 \times Y_1$, which results in $E = \{(\text{dblp:article}_5, \text{nsf:paper}_5)\}$. The similarity is now 1.00, i.e., it is very likely that link l_3 is a true positive link. The method then finds $X_2 = \{\text{dblp:affiliation}_1\}$ by following resource dblp:binliu through path $\langle \text{dblp:relatedTo} \rangle$; next, it finds $X_2 = \{\text{nsf:award}_4\}$ by following resource nsf:*binwliu* through path $\langle \text{nsf:}supports \rangle$. Now, support rule r_3 is applied and the result is intersected with $X_2 \times Y_2$, which results in $E = \{(\text{dblp:affiliation}_1, \text{nsf:award}_4)\}$. The similarity is 1.00, again this points out that l_3 is likely to be a true positive link.

Assume that $\theta = 0.50$ and $\gamma = 0.70$, which are intended for illustration purposes only. Method computeCorrespondences returns set $K = \{(\text{dblp:}weiwang, \text{nsf:}weiwang_1), (\text{dblp:binliu}, \text{nsf:}binwliu)\}$. We now have to analyse support rules r_2 and r_3 and the links that they produced given the paths shown previously. Support rule r_2 produces links l_1 and l_3, and support rule r_3 produces only link l_3; therefore, every support rule that returns at least $2\gamma = 1.40$ links taking into account the previous paths is selected. In other words, support rule r_2 is selected and support rule r_3 is discarded. The resulting context-aware rule is as follows:

$$
\begin{aligned}
r^*\text{: } & link(A, B) \text{ if } r_1(A, B), \\
& \quad X = findResources(A, \langle \text{dblp:}writtenBy \rangle), \\
& \quad Y = findResources(B, \langle \text{nsf:}leads, \text{nsf:supports} \rangle), \\
& \quad E = \{(U, V) \mid U \in X \wedge V \in Y \Rightarrow r_2(A, B)\}, \\
& \quad computeSimilarity(X, Y, E) \geq 0.50.
\end{aligned}
$$

The intuitive interpretation is that r^* links resources A and B if rule r_1 links them and they have some neighbours (using paths $\langle dblp{:}writtenBy \rangle$ and $\langle nsf{:}leads, nsf{:}supports \rangle$), respectively) that can be linked by means of rule r_2 and it results in two sets of resources whose similarity is at least 0.50.

4.3 Experimental analysis

In this section, we first describe our experimental environment and then comment on our results regarding effectiveness and efficiency. Finally, we present our statistical analysis to support our findings.

4.3.1 Experimental environment

Implementation. We implemented our proposal[†1] with Java 1.8 and the following components: Jena TDB 3.2.0 to work with RDF data, ARQ 3.2.0 to work with SPARQL queries, and Simmetrics 1.6.2, SecondString 2013-05-02, and JavaStringSimilarity 1.0.1 to compute string similarities.

Running experiments. We run our experiments in the facility described in Appendix §A.1. We used Sorbas as baseline to learn link rules relying on one sub-dataset; in addition, we run Teide in the same sub-dataset as well. Next, we applied the learnt link rules from Sorbas into a new non-overlapping sub-dataset; we ran the same rules using Teide. We aim at showing that in the first step, Sorbas takes as much time as Teide to learn context-aware link rules, nevertheless, in the second step Sorbas spends much less time than Teide to apply the link rules.

Measures. On the one hand, we explored a large portion of the parameter space to establish optimal values for θ, γ in each scenario. On the other hand, we measured the number of links returned by each proposal ($Links$), precision (P), recall (R), and the F_1 score (F_1).

Datasets sizes. We set up the following evaluation scenarios[†2]:

- In the Restaurants the training sub-datasets have 112 and 752 resources, respectively. In addition, we provided 56 $owl{:}sameAs$ links and 42,432 $owl{:}differentFrom$ links. The validation datasets have 112 and 752 resources, respectively. We provided 56 $owl{:}sameAs$ links and 42,432 $owl{:}differentFrom$ links.

[†1] The prototype is available at https://github.com/AndreaCimminoArriaga/Sorbas.

[†2] The datasets are available at http://dx.doi.org/10.5281/zenodo.2555034.

- In the scenario Publications the training sub-datasets have 108 and 98 resources, respectively. In addition, we provided 54 owl:sameAs links and 5,238 owl:differentFrom links. The validation datasets have 108 and 98 resources, respectively. We provided 54 owl:sameAs links and 5,238 owl:differentFrom links.

- In the scenario Movies the training sub-datasets have 62 and 129 resources, respectively. In addition, we provided 29 owl:sameAs links and 4,848 owl:differentFrom links. The validation datasets have 62 and 130 resources, respectively. We provided 29 owl:sameAs links and 4,848 owl:differentFrom links.

- In the scenario Films the training sub-datasets have 691 and 445 resources, respectively. In addition, we provided 222 owl:sameAs links and 153,525 owl:differentFrom links. The validation datasets have 691 and 445 resources, respectively. We provided 223 owl:sameAs links and 153,525 owl:differentFrom links.

- In the scenario Authors the training sub-datasets have 4,545 and 9,076 resources, respectively. In addition, we provided 14 owl:sameAs links and 244,689 owl:differentFrom links. The validation datasets have 512 and 707 resources, respectively. We provided 328 owl:sameAs links and 243,048 owl:differentFrom links.

- In the scenario Researchers the training sub-datasets have 130 and 101 resources, respectively. In addition, we provided 17 owl:sameAs links and 6,548 owl:differentFrom links. The validation datasets have 130 and 101 resources, respectively. We provided 16 owl:sameAs links and 6,549 owl:differentFrom links.

- In the Persons1 the training sub-datasets have 482 and 500 resources, respectively. In addition, we provided 250 owl:sameAs links and 124,750 owl:differentFrom links. The validation datasets have 484 and 500 resources, respectively. We provided 250 owl:sameAs links and 124,750 owl:differentFrom links.

- In scenario Persons2 the training sub-datasets have 528 and 400 resources, respectively. In addition, we provided 200 owl:sameAs links and 119,800 owl:differentFrom links. The validation datasets have 530 and 400 resources, respectively. We provided 200 owl:sameAs links and 119,800 owl:differentFrom links.

- In scenario Doremus16-9ht the training sub-datasets have 25 and 32 resources, respectively. In addition, we provided 16 owl:sameAs links and 496 owl:differentFrom links. The validation datasets have 25 and 32 resources, respectively. We provided 16 owl:sameAs links and 496 owl:differentFrom links.

- In scenario Doremus16-fp the sub-dataset have 29 and 42 resources, respectively. In addition, we provided 20 owl:sameAs links and 820 owl:differentFrom links. The validation datasets have 30 and 41 resources, respectively. We provided 21 owl:sameAs links and 820 owl:differentFrom links.

- In scenario Doremus17-ht the sub-dataset have 132 and 238 resources, respectively. In addition, we provided 23 owl:sameAs links and $28,300$ owl:differentFrom links. The validation datasets have 129 and 238 resources, respectively. We provided 24 owl:sameAs links and $28,297$ owl:differentFrom links.

- In scenario Doremus17-fp the sub-dataset have 75 and 75 resources, respectively. In addition, we provided 7 owl:sameAs links and $2,805$ owl:differentFrom links. The validation datasets have 75 and 75 resources, respectively. We provided 8 owl:sameAs links and $2,805$ owl:differentFrom links.

4.3.2 Experimental results

Table §4.1.(a) reports on the number of links generated by each proposal and their average precision, recall, and F_1 score using 2-fold cross validation. In the case of Sorbas, we also report on the values learnt for thresholds θ and γ.

Note that the number of links than Teide and Sorbas generated is approximately the same. The average precision of Teide and Sorbas are 0.71 ± 0.21 and 0.74 ± 0.23, respectively. Recall that we are dealing with scenarios in which the datasets have many similar resources that are actually different, and also many dissimilar resources that are actually the same. Both Teide and Sorbas take the context of the resources to be linked into account, which helps make a difference between the previous cases. Note that the recalls of Teide and Sorbas are 0.76 ± 0.21 and 0.78 ± 0.19, respectively. Realise that the F_1 score of Teide and Sorbas are 0.71 ± 0.19 and 0.75 ± 0.2, respectively. The conclusion is that both Sorbas and Teide are similar to each other regarding their effectiveness.

	Teide						Sorbas			
Scenario	Links	P	R	F_1	θ	γ	Links	P	R	F_1
Researchers	26	0.59	1.00	0.74	0.01	0.84	16	1.00	1.00	1.00
Authors	338	0.97	1.00	0.98	0.01	0.01	338	0.97	0.99	0.98
Films	232	0.96	1.00	0.98	0.01	0.85	232	0.96	1.00	0.98
Movies	32	0.50	0.55	0.52	0.01	0.01	32	0.50	0.55	0.52
Publications	44	0.64	0.39	0.48	0.01	0.01	66	0.62	0.76	0.68
Restaurants	49	0.96	0.84	0.90	0.01	0.01	49	0.96	0.84	0.90
Persons1	336	0.72	0.96	0.82	0.01	0.01	336	0.72	0.96	0.82
Persons2	195	0.69	0.67	0.68	0.01	0.01	195	0.69	0.67	0.68
Doremus16-9th	12	1.00	0.75	0.86	0.01	0.01	12	1.00	0.75	0.86
Doremus16-fp	19	0.58	0.52	0.55	0.01	0.01	18	0.56	0.48	0.51
Doremus17-ht	58	0.34	0.83	0.49	0.01	0.01	58	0.34	0.83	0.49
Doremus17-fp	7	0.57	0.57	0.57	0.01	0.01	7	0.57	0.57	0.57

a) Effectiveness results

	Teide		Sorbas	
Scenario	Learning set	Validation set	Learning set	Validation set
Researchers	17'01''	15'56''	16'15''	00'58''
Authors	08'09''	23'54''	06'11''	18'31''
Films	51'00''	48'46''	44'38''	01'57''
Movies	17'59''	13'59''	10'19''	00'27''
Publications	01'46''	01'58''	02'00''	00'09''
Restaurants	00'31''	00'32''	00'31''	00'34''
Persons1	02'11''	04'17''	02'11''	01'29''
Persons2	01'25''	01'22''	01'21''	01'22''
Doremus16-9th	00'45''	00'41''	01'07''	00'10''
Doremus16-fp	12'03''	12'49''	11'54''	00'13''
Doremus17-ht	17'19''	12'33''	23'33''	01'50''
Doremus17-fp	06'17''	06'05''	02'21''	00'12''

b) Efficiency results

Table 4.1: *Experimental effectiveness and efficiency*

Table §4.1.(b) shows our experimental results regarding efficiency. Note that Teide is an instance-driven proposal that does not attempt to learn any rules, whereas Sorbas is a rule learner. We divided our scenarios into learning and validation sets in order to perform 2-fold validation. In Table §4.1.(b), the column regarding the learning set must be interpreted as the time taken to link the resources in this set in the case of Teide and the time taken to learn context-aware rules from this set in the case of Sorbas; the column regarding the validation set must be interpreted as the time taken to link the resources in this set in the case of Teide and the time taken to apply the context-aware rules learnt previously in the case of Sorbas.

Regarding the learning set, Teide took $16'22'' \pm 23'58''$ in average and Sorbas took $16'54'' \pm 22'38''$ in average. That is, it seems that the time taken to

Precision		
Empirical rank		Iman-Davenport
Proposal	Rank	P-Value
Sorbas	1.54	0.76
Teide	1.46	

Recall		
Empirical rank		Iman-Davenport
Proposal	Rank	P-Value
Sorbas	2.33	0.88
Teide	2.21	

F_1 score		
Empirical rank		Iman-Davenport
Proposal	Rank	P-Value
Sorbas	1.42	0.52
Teide	1.58	

P-Values that are smaller than 0.05

a) Relevance of the effectiveness

Efficiency on the learning set		
Empirical rank		Iman-Davenport
Proposal	Rank	P-Value
Sorbas	1.71	2.93E-08
Teide	1.29	

Efficiency on the validation set		
Empirical rank		Iman-Davenport
Proposal	Rank	P-Value
Sorbas	1.92	5.17E-13
Teide	1.08	

P-Values that are smaller than 0.05 are greyed.

b) Relevance of the efficiency

Table 4.2: *Statistical analysis.*

link the dataset and to learn context-aware rules from it are very similar. Regarding the validation set, the difference is more clear: note that the average time taken by Teide is $16'54'' \pm 22'38''$ and the average time taken by Sorbas is $02'19'' \pm 05'09''$. That is, it seems that applying a rule that was learnt previously helps save much computing time.

4.3.3 Statistical analysis

We have confirmed our conclusion regarding the effectiveness of Sorbas by means of a statistical analysis, cf. Table §4.2.(a). Let us focus on the F_1

score since it combines precision and recall. Note that Sorbas and Teide obtained a p-value of 0.52 which is way above 0.05; the same happens for the precision p-value 0.76, and the recall p-value 0.88. These results are a strong statistical indicator that there are no relevant differences in terms of effectiveness between Sorbas and Teide.

To confirm our conclusions about the efficiency of Sorbas, we analysed the experimental results using Iman-Davenport's test, cf. Table §4.2.(b). Note that the p-value is nearly zero in both cases, which is a strong indication that the differences in rank are statistically significant. Simply put, the experimental results support the idea that Sorbas is more efficient than Teide.

4.4 Conclusions

Data inter-operability of business systems based on Web of Data requires to link the resources that are available in different datasets and represent the same real-world entities. Such links are generated by link rules that take the values of the attributes of the resources into account, but not their neighbours, which sometimes results in false positives that have a negative impact on their precision. We have presented a novel proposal that leverages a genetic programming to learn a set of link rules and then boosts them, which has proven to improve the overall F_1 score.

Chapter 5

Conclusions

Nowadays the Web of Data has become a vital resource for many companies. In order to exploit the full potential of the Web of Data it is necessary to provide an unified a linked view of their datasets, many of which are currently isolated. Link discovery is the task that aims at linking the resources in different datasets that refer to the same real-world entities. The proposals to perform link discovery must analyse a large space of potential solutions, and due to this reason, many address this problem relying on meta-heuristics; such as the genetic programming. Nevertheless, building this kind of proposal is not a trivial task, and thus, compare the current proposals under the same experimental conditions has not been done. In addition, we have proved that these rules are not as precise as they could be since they fall short when linking resources that refer to the same real-world entity but are dissimilar, and on the contrary, when resources that refer to different real-world entities are similar.

In this dissertation, we address these challenges and provide a solution. We devised a framework that allows to implement genetic programming proposals, and thus, also compare them fairly under the same experimental conditions. Relying on our framework we implemented three proposals in the literature and we devised three additional ones. Then, using well-established scenarios from the literature we ranked these proposals by means of their effectiveness. As a result, we conclude which behaves better in the different scenarios. As far as we know, the literature does not count with a clear methodology to follow when comparing the results this kind of proposals. Due to this reason, in future, we would like to propose a methodology based on hypothesis testing to compare link discovery proposals. In addition, we would like to present several heuristics that allow to reduce the search space of any link discovery proposal before it is executed, easing the computation that shall be performed by any proposal to find a suitable link rule.

On the other hand, we have analysed the precision of the link rules in scenarios where we found out that they fall short. This challenge is addressed by Teide that is a proposal that receives a set of link rules, and then, applies such rules exploiting these rules that link resources related with other resources being linked too, i.e., exploits the context of data that can be linked with the rules provided. The main drawbacks of Teide is that it requires to explore the data to find relevant resources in both data sources that can be linked with any provided rule, and also, the fact that not all the provided rules are appealing but Teide has to analyse their suitability; this leads to an expensive computational cost. To short out the complexity of Teide, we devised Sorbas that learns contextual link rules that encode the functionality achieved by Teide; in other words, Sorbas is as effective as Teide but much more efficient. Our results advocate that both Teide and Sorbas improve significantly the precision of the link rules without a significant drop of their recall. In addition, we statistically proved that Sorbas is more efficient that Teide, but their results are equivalent.

Summing up, assuming that our research hypothesis is accepted, we think that we have sufficiently proven our thesis. We hope that our results can effectively help companies to integrate data coming from sparse islands of information available on the Web. We also think that we have opened up an interesting research path that may soon lead to new research results.

Appendix A

Experimental Environment

This chapter introduces the details of the experimental environment used in our experiments. It is organised as follows: Section §A.1 describes the computing facility we used; Section §A.2 presents the different scenarios of our experiments; and finally, Section §A.3 describes the setups that we defined to run genetic programming-based link discovery proposals.

A.1 Computing facility

We run our experiments on a computer that was equipped with four Intel Xeon E5-2690 cores at 2.60 GHz and 4 GiB of RAM. The operating system was CentOS Linux 7.3.

A.2 Linking scenarios

Next, we introduce the scenarios that we used in our experiments. Most of them come from the literature and have not being curated nor modified; a few were curated or handcrafted in the context of this PhD thesis.

Researchers. This scenario relies on the 100 top authors from the Digital Bibliography & Library Project [†1] (DBLP) in 2015, and 130 researchers with the same names that were found in the National Science Foundation [†2] (NSF). The datasets and the gold standard of this scenario were created and curated by us. The Researcher scenario has been used in several articles [17–19], that have validated and refined the data and the gold standard.

Authors. This scenario consist of 9 076 authors from the Digital Bibliography & Library Project (DBLP) who share the same names, or very similar, but who are known to be different people. The datasets were created by us, however DBLP already has deduplicated all these authors, and therefore, the gold standard was extracted directly from the DBLP. The Authors scenario has been used in several articles [17–19]

Films. This scenario consist of 691 movies from the BBC and 445 films from DBpedia, having movies and films similar titles. Notice that this scenario uses a subset of the original BBC and DBpedia datasets. Due to the smaller size of these datasets, the gold standard was written by us. The Films scenario has been used in several articles [17–19]

Movies. This scenario consist of 96 movies from DBpedia and 101 films from the well-known IMDb database. The movies and films were selected by us due to the similarity of their titles. Notice that these datasets are subsets of the original ones. Thanks to their smaller size we wrote the gold standard. The Movies scenario has been used in several articles [17–19]

[†1] The dataset is available at https://dblp.uni-trier.de.

[†2] The dataset is available at https://www.nsf.gov/.

Publications. This scenario is built on top of 108 publications from the Research Assessment Exercise (RAE) and 98 papers published by the Newcastle university. The publications and papers were selected due to the similarity of their titles. The RAE and the Newcastle datasets were published by the RKB-Explorer [†3], the he gold standard of scenario was provided by the LinkLion data portal[†4].

Articles. This scenario consist of 1 600 articles from the Digital Bibliography & Library Project (DBLP) and 800 articles from the ACM Digital Library. The datasets and the gold standard were devised by Köpcke and others [35] and published in the Leipzig repository[†5].

RestaurantsZ. This scenario consist of 112 and 752 restaurants extracted from the Fodor's and Zagat's restaurant guides. The datasets of this scenario were defined in the DuDe repository[†6], that was devised by Chaudhuri and others [13]. The gold standard of this scenario was provided as well by Chaudhuri and others [13].

Restaurants. this scenario consist of 113 and 752 restaurants. The datasets and the gold standard were defined in the Ontology Alignment Evaluation Initiative (OAEI) contest of 2010[†7]; relying on the RestaurantsZ scenario. This scenario is very is well-known in the link discovery literature, and it has been used widely. It introduces noise in the data, for instance some restaurants have the same name but different telephones although they are the same, or the same address is written differently in two resources that refer to the same restaurants.

Persons1. This scenario consist of 500 and 500 resources of people whose names are very similar but are different people and vice versa. The datasets and the gold standard were defined in the Ontology Alignment Evaluation Initiative (OAEI) contents of 2010[§†7]. This scenario is very is well-known in the link discovery literature, and it has been used widely.

[†3]The dataset is available at http://www.rkbexplorer.com/data/.

[†4]The dataset is available at http://www.linklion.org/.

[†5]The dataset is available at https://dbs.uni-leipzig.de/en/research/projects/object_matching/fever/benchmark_datasets_for_entity_resolution.

[†6]The dataset is available at https://hpi.de/naumann/projects/data-quality-and-cleansing/dude-duplicate-detection.html#c114715.

[†7]The dataset is available at http://oaei.ontologymatching.org/2010.

Persons2. This scenario consist of 400 and 600 resources of different people. The datasets and the gold standard were defined in the Ontology Alignment Evaluation Initiative (OAEI) contents of 2010[§†7]. This scenario is similar to Persons1, but introduces more noise and wrong literals in the data making harder for the proposal to distinguish similar resources that refer to different people. This scenario is very is well-known in the link discovery literature, and it has been used widely.

Doremus16-9ht. This scenario consist of 40 music works that are catalogued by the French National Library and 40 music works from the Paris Philharmonic. The datasets and the gold standard were defined in the Ontology Alignment Evaluation Initiative (OAEI) contest of 2016[†8]. The very same music works of these datasets have spelling mistakes, or have some data properties mixed up on purpose. The bottom line of this scenario is to have resources with very different literals, although they refer to the same music work.

Doremus16-fp. This scenario consist of 85 music works that are catalogued by the French National Library and 41 music works from the Paris Philharmonic. The datasets and the gold standard were defined in the Ontology Alignment Evaluation Initiative (OAEI) contest of 2016[§†8]. Different music works in this datasets have very similar, or even the same, data properties. The bottom line of this scenario is to have resources that are almost the same, but refer to different music works.

Doremus17-ht. This scenario consist of 238 music works that are catalogued by the French National Library and 238 music works from the Paris Philharmonic. The datasets and the gold standard were defined in the Ontology Alignment Evaluation Initiative (OAEI) contest of 2016[†9]; extending the scenario of Doremus16-9ht. The very same music works of these datasets have spelling mistakes, or have some data properties mixed up on purpose. The bottom line of this scenario is to have resources with very different literals, although they refer to the same music work.

Doremus17-fp. This scenario consist of 75 music works that are catalogued by the French National Library and 75 music works from the Paris Philharmonic. The datasets and the gold standard were defined in the Ontology

[†8]The dataset is available at http://oaei.ontologymatching.org/2016/.

[†9]The dataset is available at http://islab.di.unimi.it/content/im_oaei/2017/.

Alignment Evaluation Initiative (OAEI) contest of 2017[§†9]; extending the scenario of Doremus16-fp. Different music works in this datasets have very similar, or even the same, data properties. The bottom line of this scenario is to have resources that are almost the same, but refer to different music works.

We distinguish two kind of scenarios from the list above, those that have been used link discovery in the literature, and those that entail a challenge for link discovery proposals due to the similarity of different resources or the noise of data that makes the same resources to have very different literals. The former kind comprises RestaurantsZ, Restaurants, Persons1, Persons2, Articles. The latter kind comprises Researchers, Authors, Films, Movies, Publications, Restaurants, Persons1, Persons2, Doremus16-9ht, Doremus16-fp, Doremus17-ht, Doremus17-fp.

A.3 Genetic programming setups

We defined 49 different setups to run our proposals. In Chapter §2, we defined a setup as a tuple (i, p, p_c, p_m, g); where i is the maximum number of iterations, p is the maximum population size, p_c is the crossover probability, p_m is the mutation probability, and g is the maximum number of generations. We gave three values to each argument: short, medium and a large. We assigned different range of values to each element in the setups following this criterion: we combined short iterations with medium and large iterations; medium iterations with short, medium and large population; and large iterations with short and medium population. Then we did the same for crossover and mutation probabilities. Finally, we combined all generated values for iterations with population and crossover with mutation probabilities. The values that arguments may take are $i = \{20, 50, 100\}, p = \{20, 100, 500\}, p_c = \{0.25, 0.50, 0.75\}$ and $p_m = \{0.25, 0.50, 0.75\}$; we set the maximum generations g to 10. Figure §A.1 recaps all the setups and the values for their arguments that we defined to run genetic programming-based link discovery proposals.

#	i	p	pc	pm	g	#	i	p	pc	pm	g
0	50	20	0.25	0.50	10	25	50	100	0.50	0.75	10
1	50	20	0.25	0.75	10	26	50	100	0.75	0.25	10
2	50	20	0.50	0.25	10	27	50	100	0.75	0.50	10
3	50	20	0.50	0.50	10	28	100	100	0.25	0.50	10
4	50	20	0.50	0.75	10	29	100	100	0.25	0.75	10
5	50	20	0.75	0.25	10	30	100	100	0.50	0.25	10
6	50	20	0.75	0.50	10	31	100	100	0.50	0.50	10
7	100	20	0.25	0.50	10	32	100	100	0.50	0.75	10
8	100	20	0.25	0.75	10	33	100	100	0.75	0.25	10
9	100	20	0.50	0.25	10	34	100	100	0.75	0.50	10
10	100	20	0.50	0.50	10	35	20	500	0.25	0.50	10
11	100	20	0.50	0.75	10	36	20	500	0.25	0.75	10
12	100	20	0.75	0.25	10	37	20	500	0.50	0.25	10
13	100	20	0.75	0.50	10	38	20	500	0.50	0.50	10
14	20	100	0.25	0.50	10	39	20	500	0.50	0.75	10
15	20	100	0.25	0.75	10	40	20	500	0.75	0.25	10
16	20	100	0.50	0.25	10	41	20	500	0.75	0.50	10
17	20	100	0.50	0.50	10	42	50	500	0.25	0.50	10
18	20	100	0.50	0.75	10	43	50	500	0.25	0.75	10
19	20	100	0.75	0.25	10	44	50	500	0.50	0.25	10
20	20	100	0.75	0.50	10	45	50	500	0.50	0.50	10
21	50	100	0.25	0.50	10	46	50	500	0.50	0.75	10
22	50	100	0.25	0.75	10	47	50	500	0.75	0.25	10
23	50	100	0.50	0.25	10	48	50	500	0.75	0.50	10
24	50	100	0.50	0.50	10						

Table A.1: *Setups defined to run proposals.*

Appendix B

Running Examples

This chapter exposes the running examples that we introduce to showcase our proposals. It is organised as follows: Section §B.1 describes the first running example based on the Researchers scenario; and Section §B.2 presents the second running example that extends the former example.

B.1 Researchers

This running example aims at illustrate a set of resources from the DBLP and NSF, which refer to people. Figure §B.1 depicts this running example, it contains three resources of type dblp:Author and three of type nsf:Researcher. The dblp:Author resources count with two data properties, i.e., dblp:name and dblp:affiliation, whereas the nsf:Researcher resources count with three data properties, i.e., nsf:name, nsf:topic, and nsf:university.

The first challenge with which every link discovery proposal must cope is aligning the data properties to be compared in order to link the different resources. In this case, the suitable data properties to be compared by a link rule are, on the one hand, dblp:name and nsf:name, and, on the other hand, dblp:affiliation and nsf:university. Regularly, comparing the names would be enough to link resources, however, it should be noticed that some NSF resources have the same name but refer to different people, i.e., nsf:$weiwang_1$ and nsf:$weiwang_2$. As a result, both pair of attributes are required to link the resources correctly.

The second challenge with which link discovery proposals must deal is the format used to encode the names of people. On the one hand, DBLP encodes the names by writing first the name and the surname, on the other hand, NSF encodes the names by writing the surname a comma and then the name. As a result, a link discovery proposal must rely on string transformations, and specifically in one that normalizes the name encodings, in order to link these resources properly.

B.2 Researchers with context

Chapters §3 and §4 explain two proposals that link instances relying on their data properties and the data properties of the resources related, i.e., the context. The running example devised in this section aims at extending the Researchers running example, by including resources as context for the resources been linked. The goal of this running example is to show of our context-based proposals work, and their related concepts.

Figure §B.2 depicts this running example, it contains two resources of type dblp:Author and three of type nsf:Researcher. The dblp:Author and the nsf:Researcher resources have only one data property, i.e., dblp:name

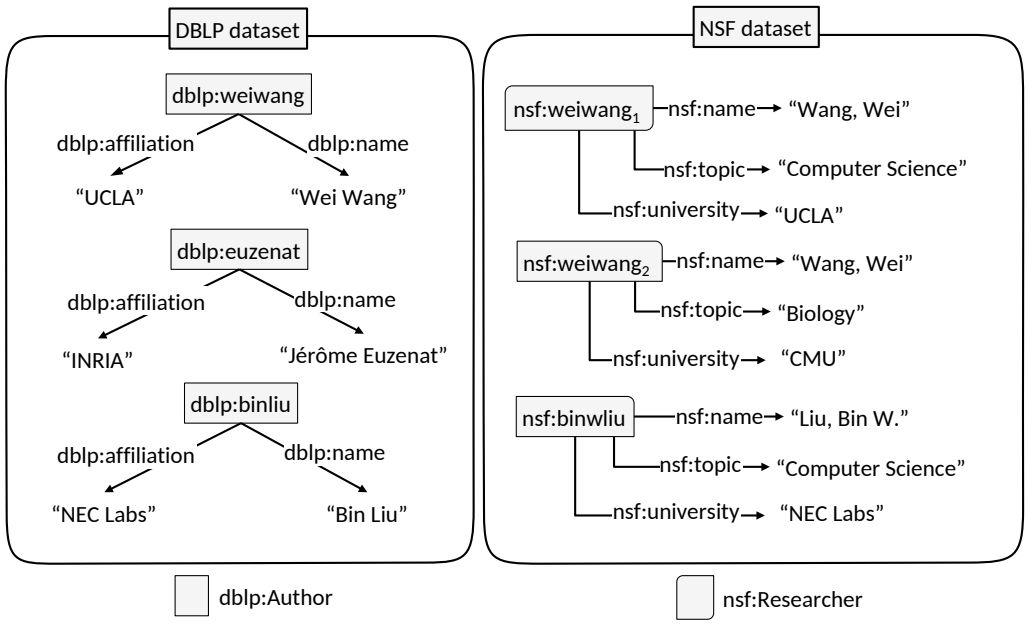


Figure B.1: *Researchers running example.*

and nsf:name. Then they have several object properties that connect them to their publications. In the case of dblp:Author the publications are directly connected, whereas the publications in the NSF are connected to the nsf:Researcher through several intermediary resources which type is nsf:Award. The DBLP and NSF publications have their titles in common

The first challenge with which a link proposals must cope is to identify suitable pairs of data properties to link instances. Resources of type dblp:Author and nsf:Researcher have a pair suitable data properties, i.e., dblp:name and nsf:name, nevertheless $nsf{:}weiwang_1$ and $nsf{:}weiwang_2$ have the same name but refer to different people. Resources of type dblp:Article and nsf:Paper have a pair of suitable data properties dblp:title and nsf:title. Finally, the resources of type dblp:Affiliation and nsf:Award have the pair of suitable attributes dblp:name and nsf:uid.

The second challenge with which a link proposals must cope is to handle the different encodings used to express the names of the resources of type dblp:Author and nsf:Researcher. As a result, link discovery proposals should count with string transformations to handle the different encodings.

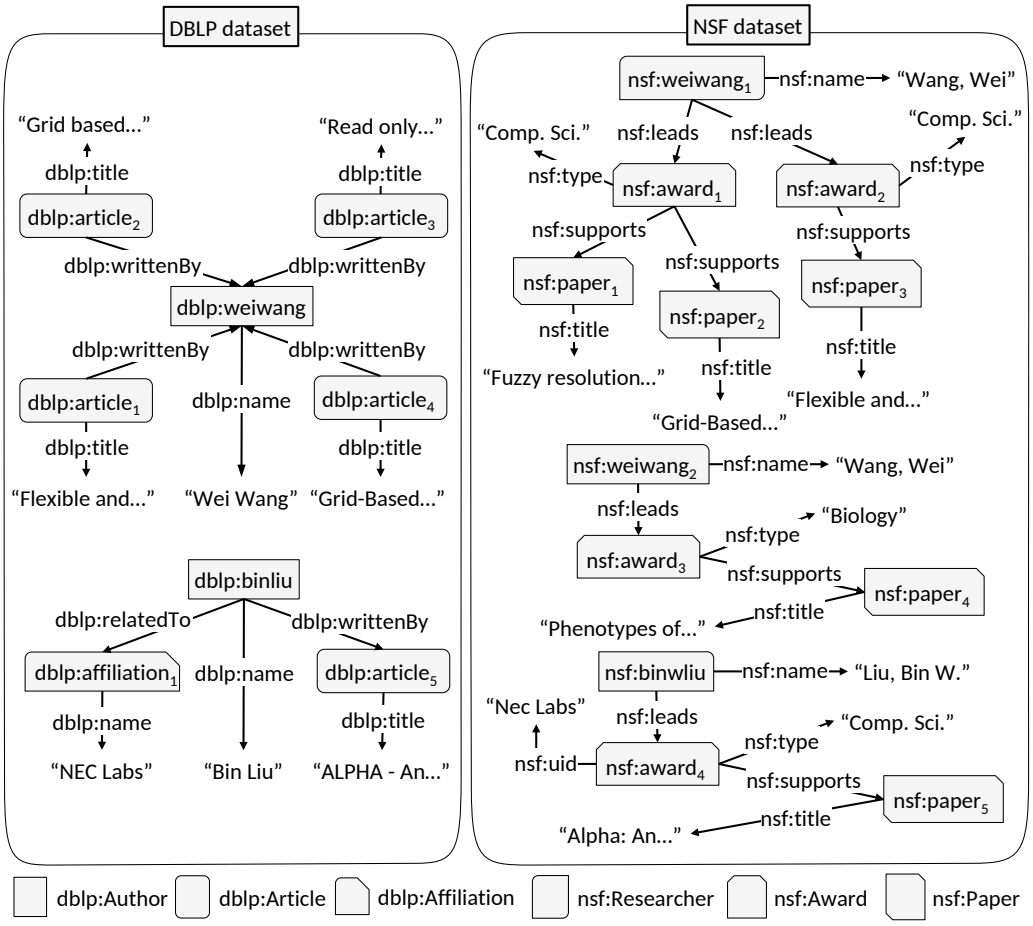


Figure B.2: *Researchers with context running example.*

Bibliography

[1] M. Achichi, M. Cheatham, Z. Dragisic, J. Euzenat, D. Faria, A. Ferrara, G. Flouris, I. Fundulaki, I. Harrow, V. Ivanova, E. Jiménez-Ruiz, K. Kolthoff, E. Kuss, P. Lambrix, H. Leopold, H. Li, C. Meilicke, M. Mohammadi, S. Montanelli, C. Pesquita, T. Saveta, P. Shvaiko, A. Splendiani, H. Stuckenschmidt, É. Thiéblin, K. Todorov, C. Trojahn, and O. Zamazal. *Results of the Ontology Alignment Evaluation Initiative 2017*. In *OM@ISWC*, pages 61–113, 2017.

[2] AEEiben and J. E. Smith. *Introduction to evolutionary computing*. Springer, 2015.

[3] H. Alili, K. Belhajjame, D. Grigori, R. Drira, and H. H. B. Ghézala. *On enriching user-centered data integration schemas in service lakes*. In *BIS*, pages 3–15, 2017.

[4] S. Araújo, J. Hidders, D. Schwabe, and A. P. de Vries. *SERIMI: resource description similarity, RDF instance matching and interlinking*. *CoRR*, abs/1107.1104, 2011.

[5] D. Beckett and B. McBride. *RDF/XML syntax specification (revised)*. Technical report, W3C, 2004.

[6] J. Belissent, E. Cullen, G. Leganza, and J. Lee. *The "data for good" movement delivers social impact*. Technical report, Forrester, 2017.

[7] J. Belissent, E. Cullen, G. Leganza, and J. Lee. *Gartner identifies top 10 data and analytics technology trends for 2019*. Technical report, Gartner, 2019.

[8] O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S. E. Whang, and J. Widom. *Swoosh: a generic approach to entity resolution*. *VLDB J.*, 18 (1):255–276, 2009.

[9] M. Bilenko and R. J. Mooney. *Adaptive duplicate detection using learnable string similarity measures*. In *KDD*, pages 39–48, 2003.

[10] C. Bizer, T. Heath, and T. Berners-Lee. *Linked Data: principles and state of the art*. In *WWW (Invited talks)*, 2008.

[11] E. N. Borges, M. G. de Carvalho, R. de Matos Galante, M. A. Gonçalves, and A. HFLaender. *An unsupervised heuristic-based approach for bibliographic metadata deduplication*. *Inf. Process. Manage.*, 47(5):706–718, 2011.

[12] S. Chaudhuri, B.-C. Chen, V. Ganti, and R. Kaushik. *Example-driven design of efficient record matching queries*. In *VLDB*, pages 327–338, 2007.

[13] S. Chaudhuri, B.-C. Chen, V. Ganti, and R. Kaushik. *DuDe: the duplicate detection toolkit*. In *VLDB*, page #5, 2010.

[14] Z. Chen, D. V. Kalashnikov, and S. Mehrotra. *Exploiting context analysis for combining multiple entity resolution systems*. In *SIGMOD Conference*, pages 207–218, 2009.

[15] P. Christen. *FEBRL: an open source data cleaning, deduplication and record linkage system with a graphical user interface*. In *KDD*, pages 1065–1068, 2008.

[16] P. Christen. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer, 2012.

[17] A. Cimmino and R. Corchuelo. *A hybrid genetic-bootstrapping approach to link resources in the Web of Data*. In *HAIS*, pages 145–157, 2018.

[18] A. Cimmino and R. Corchuelo. *On feeding business systems with linked resources from the Web of Data*. In *BIS*, pages 307–320, 2018.

[19] A. Cimmino, C. R. Rivero, and D. Ruiz. *Improving link specifications using context-aware information*. In *LDOW*, 2016.

[20] I. F. Cruz, F. P. Antonelli, and C. Stroe. *AgreementMaker: efficient matching for large real-world schemas and ontologies*. *PVLDB*, 2(2): 1586–1589, 2009.

[21] M. G. de Carvalho, A. HFLaender, M. A. Gonçalves, and A. S. da Silva. *A genetic programming approach to record deduplication. IEEE Trans. Knowl. Data Eng.*, 24(3):399–412, 2012.

[22] M. G. de Carvalho, A. HFLaender, M. A. Gonçalves, and T. C. Porto. *The impact of parameters setup on a genetic programming approach to record deduplication.* In *SBBD*, pages 91–105, 2008.

[23] M. G. Elfeky, A. K. Elmagarmid, and V. S. Verykios. *TAILOR: a record linkage tool box.* In *ICDE*, pages 17–28, 2002.

[24] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. *Duplicate record detection: a survey. IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.

[25] J. Freitas, G. L. Pappa, A. S. da Silva, M. A. Gonçalves, E. S. Moura, A. Veloso, A. H. Laender, and M. G. de Carvalho. *Active learning genetic programming for record deduplication.* In *IEEE Congress on Evolutionary Computation*, pages 1–8, 2010.

[26] R. García-Castro and A. Gómez-Pérez. *Guidelines for benchmarking the performance of ontology management APIs.* In *ISWC*, pages 277–292, 2005.

[27] J. Huber, T. Sztyler, J. Nößner, and C. Meilicke. *CODI: combinatorial optimization for data integration.* In *OM*, pages 134–141, 2011.

[28] R. Isele and C. Bizer. *Learning expressive linkage rules using genetic programming. PVLDB*, 5(11):1638–1649, 2012.

[29] R. Isele, A. Jentzsch, and C. Bizer. *Active learning of expressive linkage rules for the Web of Data.* In *ICWE*, pages 411–418, 2012.

[30] V. Jahns. *Principles of data integration by Anhai Doan, Alon Halevy, Zachary Ives. ACM SIGSOFT Software Engineering Notes*, 37(5):43, 2012.

[31] E. Jiménez-Ruiz and B. C. Grau. *LogMap: logic-based and scalable ontology matching.* In *International Semantic Web Conference*, pages 273–288, 2011.

[32] H. Köpcke and E. Rahm. *Training selection for tuning entity matching.* In *QDB/MUD*, pages 3–12, 2008.

[33] H. Köpcke and E. Rahm. *Frameworks for entity matching: a comparison. Data Knowl. Eng.*, 69(2):197–210, 2010.

[34] H. Köpcke, A. Thor, and E. Rahm. *Comparative evaluation of entity resolution approaches with FEVER*. *PVLDB*, 2(2):1574–1577, 2009.

[35] H. Köpcke, A. Thor, and E. Rahm. *Evaluation of entity resolution approaches on real-world match problems*. *PVLDB*, 3(1):484–493, 2010.

[36] H. Köpcke, A. Thor, and E. Rahm. *Learning-based approaches for matching web data entities*. *IEEE Internet Computing*, 14(4):23–31, 2010.

[37] L. Leitão, P. Calado, and M. Weis. *Structure-based inference of XML similarity for fuzzy duplicate detection*. In *CIKM*, pages 293–302, 2007.

[38] Z. Michalewicz. *Evolutionary programming and genetic programming*. In *Genetic Algorithms + Data Structures = Evolution Programs*, pages 283–287. Springer, 1996.

[39] M. Nentwig, M. Hartung, A.-C. N. Ngomo, and E. Rahm. *A survey of current link discovery frameworks*. *Semantic Web*, 8(3):419–436, 2017.

[40] A.-C. N. Ngomo and S. Auer. *LIMES: a time-efficient approach for large-scale link discovery on the Web of Data*. In *IJCAI*, pages 2312–2317, 2011.

[41] A.-C. N. Ngomo and K. Lyko. *EAGLE: efficient active learning of link specifications using genetic programming*. In *ESWC*, pages 149–163, 2012.

[42] A.-C. N. Ngomo and K. Lyko. *Unsupervised learning of link specifications: deterministic vs. non-deterministic*. In *OM*, pages 25–36, 2013.

[43] K. Nguyen, R. Ichise, and B. Le. *SLINT: a schema-independent linked data interlinking system*. In *OM*, 2012.

[44] A. Nikolov, M. d'Aquin, and E. Motta. *Unsupervised learning of link discovery configuration*. In *ESWC*, pages 119–133, 2012.

[45] A. Nikolov, V. S. Uren, and E. Motta. *KnoFuss: a comprehensive architecture for knowledge fusion*. In *K-CAP*, pages 185–186, 2007.

[46] A. Nikolov, V. S. Uren, E. Motta, and A. N. de Roeck. *Integration of semantically annotated data by the KnoFuss architecture*. In *EKAW*, pages 265–274, 2008.

[47] X. Niu, S. Rong, Y. Zhang, and H. Wang. *Zhishi.links results for oaei 2011*. In *OM*, 2011.

[48] R. Poli, W. B. Langdon, and N. F. McPhee. *A field guide to genetic programming*. lulu.com, 2008.

[49] A. Singh and A. Sharan. *Adaptive genetic programming based linkage rule miner for entity linking in Semantic Web*. In *ICCCA*, pages 373–378, 2017.

[50] T. Soru and A.-C. N. Ngomo. *A comparison of supervised learning classifiers for link discovery*. In *SEMANTICS*, pages 41–44, 2014.

[51] C. Sun, D. Shen, Y. Kou, T. Nie, and G. Yu. *ERGP: a combined entity resolution approach with genetic programming*. In *IEEE WISA*, pages 215–220, 2014.

[52] J. Tang, B. Liang, J.-Z. Li, and K. Wang. *Risk minimization based ontology mapping*. In *AWCC*, pages 469–480, 2004.

[53] S. Tejada, C. A. Knoblock, and S. Minton. *Learning object identification rules for information integration. Inf. Syst.*, 26(8):607–633, 2001.

[54] A. Thor and E. Rahm. *MOMA: a mapping-based object matching system*. In *CIDR*, pages 247–258, 2007.

[55] S. I. Vargas, G. O'Donnell, W. McKeon-White, and D. Lynch. *From an island to a web: the modern data center*. Technical report, Forrester, 2018.

[56] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. *Silk: a link discovery framework for the Web of Data*. In *LDOW*, 2009.

[57] N. Yuhanna, G. Leganza, and E. Hoberman. *The Forrester Wave: big data fabric*. Technical report, Forrester, 2018.

[58] H. Zhao and S. Ram. *Entity identification for heterogeneous database integration. Inf. Syst.*, 30(2):119–132, 2005.

This document was typeset on July 20, 2019 at 19:55 using class RG-BOOK α2.14 for LaTeX$_{2\varepsilon}$. As of the time of writing this document, this class is not publicly available since it is in alpha version. Only members of The Distributed Group are using it to typeset their documents. Should you be interested in giving forthcoming public versions a try, please, do contact us at contact@tdg-seville.info. Thanks!