Semantic enrichment of data: annotation and data linking

Nathalie Pernelle

Université Paris Sud
Laboratoire de Recherche en Informatique

Mémoire

présenté en vu d'obtenir l' Habilitation à Diriger des Recherches,
spécialité "Informatique"

par

Nathalie Pernelle

Semantic enrichment of data: annotation and data linking

HDR soutenue le le 27 juin 2016 devant le jury composé de :

M^me	Chantal Reynaud	Professeur, Université Paris Sud	(Marraine)
M.	Mathieu D'Aquin	Senior Researcher, The Open University	(Rapporteur)
M^me	Marie-Laure Mugnier	Professeur, Université de Montpellier	(Rapporteur)
M.	Yannick Toussaint	CR (HDR), Université de Nancy	(Rapporteur)
M.	Patrice Buche	IR (HDR) INRA	(Examinateur)
M.	Thierry Charnois	Professeur, Université Paris 13	(Examinateur)
M^me	Christine Froidevaux	Professeur, Universite Paris Sud	(Examinateur)

Remerciements

Je voudrais tout d'abord adresser mes plus sincères remerciements aux membres de mon jury qui m'ont fait l'honneur d'accepter de lire ce manuscrit et de m'écouter. Je remercie en particulier Mathieu D'Aquin, Marie-Laure Mugnier et Yannick Toussaint pour avoir accepté la charge de rapporter sur mon habilitation. Merci, vraiment. Je remercie de tout coeur Christine Froidevaux d'avoir accepté de présider ce jury.
J'adresse mes plus chaleureux remerciements à Chantal Reynaud, sans qui je n'aurais pas fait cette démarche. Je la remercie pour sa disponibilité et pour sa bienveillance : être un chercheur brillant et généreux, ce n'est pas donné à tout le monde !

Il y a certainement des chercheurs qui préfèrent effectuer leurs recherches seuls, mais soyons clairs, ce n'est pas mon cas. J'ai en particulier adoré travailler toutes ces années avec Fatiha Sais qui résiste encore à mes conversations parfois étranges. Merci Fatiha pour ton amitié et ton optimisme si précieux. Je souhaite également remercier mes (autres) étudiants en thèses, Mouhamadou Thiam, Yassine Mrabet, Danai Symeonidou et Joe Raad, tous différents mais tous enthousiastes. Danai, j'espére que tu sauras garder ton énergie si constructive et si communicative toujours intacte ! Je tiens également à remercier les chercheurs avec qui j'ai eu la chance de collaborer. Je pense en particulier à Laura Papaleo, Françoise Gayral, Henri Soldano, Juliette Dibie, Ollivier Haemmerlé, Liliana Ibanescu, ou encore Vincent Armant, merci pour tout ce que vous m'avez appris.

Je remercie également les membres et anciens membres de mon équipe avec qui j'ai effectué ce bout de chemin. Chère Brigitte, je te ramène quand tu veux, c'est toujours un plaisir ... Haifa, Gloria, Hélène, vos rires me manquent. Bon, Sarah, tu es dans l'équipe Bioinfo mais j'ai vraiment de la chance que tu ne sois pas loin, merci pour tes conseils.

J'en profite également pour remercier mes anciens collègues du LIPN à Paris 13 et en particulier Nathalie Chaignaud, Nathalie Bouquet, et Sylvie Szulman, merci pour tout ce que vous m'avez apporté et pour votre bonne humeur.

Un grand merci à Martine Lelièvre, Sylvie Menou et Stéphanie Druetta qui savent rester imperturbables quels que soient les problèmes administratifs rencontrés.

Merci à mes collègues de l'IUT de Sceaux ou de l'EISTI, en particulier à ceux avec qui j'ai eu la chance de travailler : Philippe Thouard, Marc Lanniaux, Isabelle Morin, Liliane Alfonsi, Chris Baskiotis et Hervé de Milleville. Philippe, tu m'impressionneras toujours !

J'aurais un mot particulier pour Daniel Kayser et Marie-Christine Rousset, qui ont tous les deux joué un rôle central dans ma carrière et pour qui j'ai

une estime incommensurable.

Je conclurai en remerciant de tout cœur Olivier et Corentin ainsi que ma flopée de frères et soeurs. Merci pour pleins de détails et merci pour pleins de choses importantes.

Contents

List of Figures

Introduction

1

This Habilitation thesis outlines some of my main research activities carried out as an Associate Professor at Université Paris Sud since I have defended my PhD at Université Paris 13 in 1999. My PhD thesis was about Natural Language Processing. In this thesis, I have proposed a logical approach to automatically discover the meaning of french expressions in which a particular kind of nominal polysemy may occur. This work shown how important the understanding of the context of words is, and how useful it is to combine the meaning conveyed by words and that suggested by the context. This work aroused my interest in data semantics. This may explain why, after a two years position at the engineering school EISTI of Cergy Pontoise, I began to work on approaches that can help users to enrich semantically more or less structured data (HTML, XML or RDF data). These research activities have been conducted in the LRI laboratory (Laboratoire de Recherche en Informatique) of Université Paris Sud.

1.1 Context

Structured data can be seen as data that are represented using a specified model. In a relational database or an XML document, the data that describes a person can be splited into last name, first name, phone number and address and conform to a relational schema or a DTD. Standard languages have been defined to describe structured resources on the web. RDF (Resource Description Framework) provides a simple language for describing information about Web resources identified by Universal Resource Identifiers (URI). In RDF, a fact about a resource is a triplet made of a subject, a predicate and an object: <s p o> expresses that the subject s has the value o for the property p. Besides, the data model can be declared using a formalized ontology. Ontologies are represented as hierarchies of concepts, properties and individuals which usually describe domain-specific knowledge. When Web Ontology Languages (OWL) are used to represent the ontology, the domain is described on the basis of description logics which formalize the classes, their properties, and the axioms and this formalization authorises reasoning on the data and the ontology.
The Linked Open Data initiative brought more and more RDF data sources to be published on the Web. In short, that seems a marvellous framework to share data and to exploit local and external data sources intelligently. Nevertheless, data sources are often heterogeneous since people organize data using different ontologies, different URIs and different ways to express the same litteral values, even in the same application domain. Linked

Data is based on the idea that even if data sources are heterogeneous, when RDF data provided by different sources are connected by semantic links, it enables softwares to explore the data and combine information published in these different sources. However, when ontologies and datasets are voluminous, all the semantic links between ontology concepts or properties and all the identity links between data items cannot be speficied manually. Various ontology alignment tools have been proposed to discover mappings between concepts or properties of distinct ontologies (Shvaiko and Euzenat (2013)). These approaches exploit terminological information, the structure of the ontology, axioms or even individuals to discover these mappings (semi)-automatically. Good results can be obtained as it has been shown in the Ontology Alignment Evaluation Initiative (OAEI) (http://oaei.ontologymatching.org) even if the results that can be generated by a given approach may vary depending on ontology characteristics (e.g. multilinguism, size, expressiveness, existence of individuals, ...). In addition, data linking approaches have focused on the discovery of identity links between individuals so that a user or an application knows that two URIs refer to the same real worl object (Ferrara et al. (2013)).

Now, obviously, many valuable information are still described in unstructured documents. Unstructured data is all the data that does not conform to a pre-defined data model : HTML Web pages, spreadsheets, raw texts, images, videos. If we focus on the textual parts of these kinds of documents, we know that interpreting natural language is a difficult task and requires many lexical, grammatical and domain knowledge in particular to discover what is not directly expressed by the literal sense of the words. Many existing tools for unstructured content description and query processing are based on keywords. These tools have limited capabilities to represent and reason about the concepts or the real world objects that could be associated with document contents: a document that talks about *asthma* talks about *lung disease*, a document that talks about *Michelle Obama* talks about the wife of *Barack Obama*. Aiming to solve the limitations of keyword-based approaches, many ontology-based approaches have investigated the issue of (semi)-automatic annotation of documents. An annotation is a metadata which is associated to a part of a document (e.g. a named entity, a nominal group, a paragraph, the whole document). A semantic annotation is an annotation such that its semantics is defined in an ontology. For example, an annotation can associate the named entity *Université Paris Sud* that occurs in a document to an ontology, identifying it as an instance of the concept $University$. Parts of the document can be annotated by concepts, such as $University$, properties such as $part - Of$, concept instances, such as *Ada Lovelace* or property values *Ada Lovelace is-born-in London*. These approaches belong to complementary research fields such as knowledge engineering, machine learning or natural language processing. Annotations can be performed semi-automatically, i.e. proposed to human experts, or can be fully automatic. The difficulties are manifold: segmentation, syntactical variations, anaphora resolution, implicit information, interpretations that involve several sentences or documents, concept or property disambiguation, or instance disambiguation (e.g. do *Paris* refer to the capital of France, a city of Texas, or the legendary son of Priam?).

HTML was designed for the visual presentation of documents on the web. It was not created to help systems to access easily to the data described in documents. However, sometimes, a web document or a part of a web document has a rather regular structure that can be taken into account in an extraction or an annotation process. For example, successful approaches have taken benefit from Wikipedia infoboxes to construct large knowledge bases (Suchanek et al. (2007), Auer et al. (2007)). Likewise, some of the unstructured data are represented in tables (spreadsheets, parts of Pdf or HTML documents) and in many domains, these tables represent synthetic data that can be of great value.

1.2 Research topics and contributions

Over the last fifteen years, I have focused on issues related to the general problem of enriching unstructured documents or structured RDF data with semantic information. This has been done following three main research axes.

- A first research axis has been dedicated to the definition of approaches that automatically construct class hierarchies over XML data when there is no domain ontology available. Since this work is rather old now, I will only describe it briefly in this chapter but I will not detail it in this thesis.
- In a second research axis, I have investigated different approaches that can be used to annotate unstructured data using available ontologies.
- A third research axis has been devoted to the enrichment of RDF datasets with identity links when RDF datasets conform to OWL ontologies.

1.2.1 Automatic construction of class hierarchies over XML data

XML can be considered as a semi-structured data-model that allows to describe the content of a document in order to facilitate data access and data exchange. However, when data descriptions are not grouped in classes that are hierarchically organized, it can be difficult for a user to access to them, in particular when these data are voluminous. In some applications, a pre-defined ontology that could be used to (semi)-automatically type the data is not available. The question is then: how can an approach automatically construct a set of hierarchically organized classes from a set of XML data?

In 2000, in the setting of the RNRT pre-competitif GAEL Project, we were interested in the construction of electronic catalogs that can help a user to access to some company products thanks to a class hierarchy. In this project, we have worked on a set of electronic products that were described using XML. Products were described by a basic type and varied sets of attributes that can be multi-valued. The aim was to construct class hierarchies and to provide an understandable description of the generated classes.

We have defined an approach, named Zoom, that can help users to access to a large amount of data by clustering them in a small number of classes described at different levels of abstraction (Pernelle et al. (2001;

2002)). Our basic representation was a Galois lattice which is a well-defined and exhaustive representation of the classes embedded in a data set. More precisely, a Galois lattice is a lattice in which terms of a representational language are partitioned into equivalence classes w.r.t their *extent* (i.e. the set of instances that satisfy the term). Each node of a Galois lattice corresponds to a *concept* represented as its extent and its *intent*. The *intent* of a concept is the most specific term corresponding to the concept extent, and so is the representative of one equivalence class of terms. One of the main drawback of Galois lattice structures is that the size of such a lattice can be very large when dealing with real-world data sets. At that time, various methods were already proposed to reduce its size by eliminating part of the nodes (Godin et al. (1995), Hereth et al. (2000), Brézellec and Soldano (1998)).

We addressed this problem by taking into account the limitations of a user (what he can take in) and his wishes (what he is interested in). The system ZooM first builds a coarse lattice covering the whole set of instances and then refines a small subpart of the coarse lattice delimited by two nodes chosen by a user. This approach uses two languages of description of classes and different extension functions. The first language of classes has a high power of abstraction and guides the construction of a lattice of classes covering the whole set of data. The idea was to exploit this language to distinguish first products thanks to the set of attributes that are used to describe them. For instance, products that are described using a property $memory - size$ are distinguished from products for which this attribute is not valued. The second language of classes, more expressive, is the basis for the refinement of a part of the lattice that the user wants to focus on. In this second language, attribute values are considered and the $memory - size$ value can be used to distinguish the different products. Furthermore, the clustering was first performed on the set of basic types instead of the data set and a more fine-grained notion of extension ($\alpha - extension$) can be used to deal with exceptional products. From a theoretical point of view, we have proposed the general framework of *nested Galois lattices* which formalizes the links between the different lattices constructed by ZooM and we have shown that nesting is based on extensional and/or an intensional projections.

This work has shown that Galois lattices is a formalization that can be relevantly used to construct hierarchically ordered concepts when data, described in a multivalued context, are associated to first types. Indeed, both the granularity level of the language used to define concept intents and the function used to compute concept extents can be used to abstract or refine the constructed hierarchy. This way, the concept hierarchy can be adapted to the user needs.

1.2.2 Semantic annotation of web documents

When ontologies are available, they can be used to semantically annotate web documents. I have investigated ontology-based approaches that are also guided by the syntactic structure of (part of) web documents.

First, in the setting of the e.dot project, we have investigated the semantic annotation of tabular data. Annotation tools that focus on tabular data can take benefit of the structure of the tables. Simple but helpful hypotheses can help to define such approaches: data of the same columns usually refer to

the same class, a semantic link usually exist between data described in the same line (Limaye et al. (2010), Zhang (2014), Cafarella et al. (2008)).

The aim the e.dot project was the automatic construction of domain specific data warehouses. The availability of scientific data on the web is a great opportunity for scientists to collect and analyse data produced by other scientists. Tools are needed to help scientists to discover, query and eventually merge such heterogeneous data. In the setting of the e.dot project, the application domain was the microbiological risks in food products. An existing system named MIEL++ (Buche et al. (2004)) was based on a database containing experimental and industrial results about the behavior of pathogenic germs in food products. In this project, methods were needed to semantically annotate tabular data extracted from scientifc articles crawled on the Web. Thus these data could be queried through a mediated architecture based on an available domain ontology. Since the system was dedicated to a scientific use, the idea was to semantically annotate as many information as possible while allowing the scientists to have access to elements of the original table in order to limit the errors due to a bad interpretation in the annotation process. We have defined a Document Type Definition named SML (Semantic Markup Language) which can deal with additional or incomplete information in a semantic relation, ambiguities or possible interpretation errors and a flexible mapping method, named Xtab2SML, that can be used to represent extracted tables using this DTD (Gagliardi et al. (2005a;b)).

Other regularities in syntactic structures of web documents can also be used to enrich ontologies or knowledge bases (populated ontologies). The aim of the ontology-based approaches developped in the SHIRI project was to exploit the syntactic structure of heterogeneous HTML documents in an annotation process. In the approaches we have defined, the HTML structure is either used to better rank the annotations when they appear in the most structured parts of the documents (Mrabet et al. (2010)) or to propose semantic relations that are difficult to discover with lexico-syntactic patterns (Mrabet et al. (2012; 2014)). Indeed, while noticeable advances are achieved for the extraction of concept instances, the extraction of semantic relations is still challenging when the structures and the vocabularies of the target documents are heterogeneous.

These works are presented in chapter 2.

1.2.3 Data Linkage

In this last topic, I have worked on the data linkage problem. To combine and reason about data coming from different RDF data sources, semantic links are needed to connect resources. In particular, identity links allow to declare that two data items refer to the same real world object, i.e that two RDF descriptions refer to the same hotel, the same lab, etc. .... Based on these links, it is possible to combine information about the same real-world entity which is stored in several repositories. This problem appears when different data sources are used but also when a user or an application wants to detect duplicates in one data source.

A first work was initially motivated by the data integration task defined in the industrial project PICSEL 3 (Production d'Interfaces à base de Connaissances pour des Services En Ligne), a project realized in collaboration

with France Telecom R&D (2005-2007). In this project, a mediation architecture was also integrating a local data warehouse which contains RDF data given by different content providers. These data were conform to a mediator schema represented in RDFS+ language (RDFS extended by a fragment of OWL-DL) and the local data warehouse was progressively enriched by external data. One of the goals of PICSEL 3 was to eliminate duplicates in the datawarehouse. At that time, many automatic approaches were existing in the relational setting (Winkler (2006a)) while very few were developed for RDF datasets which conform to OWL ontologies (Dong et al. (2005)). We have defined two approaches which exploit knowledge that can be declared in the ontology (Saïs et al. (2007; 2009; 2010)). In a first logical approach named L2R, some ontology axioms (disjunctions, inverse functional properties, composite keys) and additionnal knowledge about the data sources are automatically translated into Horn rules and used to infer (non) reconciliations between reference pairs and (non) synonymies between values (Saïs et al. (2007)). In a second approach, knowledge semantics is automatically translated into non-linear equations which allow to compute similarity scores for reference pairs (Saïs et al. (2009; 2010)). We have also proposed an approach based on colored petri-nets to graphically explain to experts the results that can be obtained with such a global and iterative method (Gahbiche et al. (2010a)).
Since this period, many other approaches have been developed to generate identity links (semi)-automatically and these approaches can be distinguished according to various criteria (Ferrara et al. (2013)). Many of them are based on discriminative properties. In OWL2, composite keys can be declared that uniquely identify data items but these keys are rarely expressed and composite keys are difficult to determine even for domain experts. In the setting of the Qualinca project, we have developped approaches that can discover simple and composite keys from RDF data sources. A first approach named KD2R, exploits data sources for which the Unique Name Assumption is stated (Symeonidou et al. (2011), Pernelle et al. (2013b)). The second approach, named SAKey, can discover keys when datasets may contain duplicates or erroneous property values (Symeonidou et al. (2014)).

Finally, automatic data linking tools may generate some wrong identity links. So, we have investigated how invalid sameAs links can be logically detected (Papaleo et al. (2014)).

These works are presented in chapter 3.

Semantic annotation 2

2.1 Introduction

In an ideal world, the web would be a vast platform that can be easily exploited by applications to integrate and query data coming from different data sources. With the linked data initiative, many semantic Web projects aim to publish structured data on the Web and to link them with existing knowledge bases. Such initiatives allow semantic search engines to query different knowledge bases and to perform reasoning tasks on their data.

However, these published data sources contain much less information than the unstructured documents available on the Web. The web is still mainly document-oriented. With classical search engines, unstructured web documents can be searched using more or less advanced keyword-based techniques: the user gives a set of terms, eventually combined using boolean expressions, and the search engine returns a set of documents that includes this given set of keywords. These kinds of approaches are limited because they cannot distinguish between different interpretations the terms may have in the documents.

Semantic annotations can be used to facilitate unambiguous access to document content. *Annotation* is about attaching an additional information to a document or to a selected part of a document. A *semantic annotation* allows to associate some ontology-based metadata to the selected part of a document. Since ontologies are formal conceptualizations of an application domain, they can provide a formalized representation of concepts, properties and instances that natural language expressions can refer to. An example annotation would associate the text *Montpellier* to the concept City described in an ontology, while a more accurate annotation would relate it to the City instance named *Montpellier* that is located in the south of France if this instance already exists in a populated ontology.

Some annotation tools allow users to add annotations to web resources, and eventually share them with others (Heflin et al. (2003), Ciccarese et al. (2012)). However, manual annotation is an expensive process. Automation is a key factor in order to exploit the contents of documents at Web scale. Diverse techniques can be used and combined to automatically annotate Web documents, ranging from natural language processing and machine learning to knowledge management.

Many research projects have been developed to extract knowledge from unstructured texts and eventually annotate them automatically. Many of them focus on extraction or annotation of named entities (Nadeau and Sekine (2007), Mendes et al. (2011), Yosef et al. (2011)). Named entities are

sequences of words in a text which are either the names of things, such as cities, person names, company names, or dates and numbers. Named entity recognition (identifying named entity in a text and assigning a class label to it) or named entity resolution (disambiguisation of the named entity according to a knowledge base) are important tasks since many facts that can be discovered in texts rely on named entities. Some approaches also aim to discover property assertions from unstructured texts (Suchanek et al. (2009; 2006), Borislav et al. (2004), Cimiano et al. (2005)). Most of these approaches (learn and) exploit lexico-syntactic patterns representing regular expressions appearing in the texts, and/or knowledge bases or (onto-)lexical resources. However, semantic annotation of texts is a difficult task for many reasons: named entities are polysemous (there are many different instances that are named *Paris*), a named entity can be imbricated in another named entity (the named entity *University Gaston Berger* contains the named entity *Gaston Berger*), coreferences need to be solved (in *He was born in december 1982*, *he* refers to a named entity previously cited in the text), semantic relations can be expressed by many surface forms (*is located in*, *is situated in* can both express a property *LocatedIn* of an ontology).

Other extraction or annotation approaches exploit the syntactic structure of web documents. The semantic annotation approaches we have defined are in line with these kinds of approaches. Indeed, even if HTML documents are formatted only for visual purposes, the underlying semantics that can be expressed by the syntactic structure can be exploited to semantically annotate web documents. Some approaches exploit structural regularities of template-based HTML documents, i.e. documents that are generated by populating templates from databases (e.g., product portals, news) (Crescenzi et al. (2002), Arasu and Garcia-Molina (2003)). Other approaches focus on the most syntactically structured parts of heterogeneous web documents. For example, approaches like DBpedia (Auer et al. (2007)) and Yago (Suchanek et al. (2007)) have constructed a knowledge base from the infoboxes of the partially structured resource Wikipedia. Other approaches are dedicated to web tables (Cafarella et al. (2008), Limaye et al. (2010), Zhang (2014), Bhagavatula et al. (2015)). Indeed, recovering semantics from web tables is an important task since more and more high-quality information are represented in tables. Furthermore, they cover many different topics. Nevertheless, this rich source of syntactically structured information cannot be so easily exploited. First, there is no explicit formal schema defined on such tables: row, column headers and cell values do not use a controlled vocabulary. Second, tables can be embedded in free text that may contain information that would help to interpret the table content. Last, although a table with a simple row and column structure is common, tables can be much more complex (e.g. nested tables, lines that correspond to aggregated values ...).

The works I present in this chapter aim to take advantage of the syntactic structure of heterogeneous documents to annotate them semantically. I first present an approach that can be used to represent web tables using a controlled vocabulary defined in a term taxonomy. Then I will present two approaches that have been defined in the setting of the SHIRI project: SHIRI-Querying, an approach that aims to annotate HTML document nodes, and

REISA, a method that enriches a populated ontology with property instances that can be discovered using neighbour nodes of HTML documents.

2.2 Semantic representation of tabular data

2.2.1 Context

I have worked on the semantic representation of tables extracted from the web from 2004 to 2005 with Fatiha Saïs (PHD student co-advised with Marie-Christine Rousset), Helène Gagliardi (MCF, Univ. Paris-Sud) and Ollivier Haemmerlé (MCF, INA-PG/INRA). This work takes place within the e.dot RNTL project, a common project between the INA P-G/INRA BIA group, the Xyleme start-up, University Paris-sud (LRI) and the Verso team (INRIA), funded by a French National Program, the *Reseau National des Technologies Logicielles*, in short RNTL. The application domain of e.dot is the food risk assessment which is an industrial and public health stake.
The proposed approach has been first described in 2005 in the french conference Extraction et Gestion de Connaissances (EGC) (Saïs et al. (2005)), in the Workshop Context and Ontology co-located with the international conference AAAI (Gagliardi et al. (2005a)) and in the international conference Discovery in Science (DS) (Gagliardi et al. (2005b)).

2.2.2 Motivation

The e.dot project deals with the automatic construction of domain specific data warehouses. The application domain concerns microbiological risks in food products. The MIEL++ system (Buche et al. (2004)), developed by INRA partners, was an existing tool based on a database containing experimental and industrial results about the behavior of pathogenic germs in food products. This database was incomplete by nature since the number of possible experiments is potentially infinite. We aim to remedy that incompleteness by complementing the database with data automatically extracted from the Web. We were interested in data tables since they are very common presentation schemes to describe synthetic data in scientific articles.

At this time, in 2004, there were not so many automatic tools that were extracting knowledge from web tables. Contrarily to previous approaches like Crescenzi et al. (2002), Arasu and Garcia-Molina (2003), our approach cannot rely on a common structure that could be discovered among a set of template-based documents. Indeed, in the setting of the e.dot project, the web tables that can be exploited to fill the data warehouse were very heterogeneous. Some works like Kushmerick (2000), Muslea et al. (2001) and Hsu and Dung (1998) allowed to extract knowledge by learning rules from a sample of manually annotated documents. Our goal was quite different since we wanted our approach to be completely automatic. More generally, recovering the semantics of a web table can be seen as a particular schema matching problem. Indeed, it implies (1) finding correspondences between table columns and ontology classes or property values, (2) identifying the set of relations that are expressed in the table. In this schema matching problem, terminological and structural similarities that exploit the title of the tables, the title of the columns and the cell values can be used. Many existing

approaches were discovering schema mappings for relational databases or XML (Rahm and Bernstein (2001), Doan et al. (2003)). These techniques based on cell values can obtain good results.

In the context of the project edot, we were inspired by these approaches but we have relied on a domain ontology (and not on a relational schema). Furthermore, in a scientific domain such as microbiological risk in food products, an automatic approach will rarely lead to a perfect and complete ontology-based representation of a table. First, several possible mappings can be discovered for one cell value of a table. For instance, in Figure 2.1, the cell value *Whiting with lemon* can be mapped to several ontology terms such as *green lemon* or *whiting fillets* with different confidence degrees depending on the used similarity measures. Second, some columns can be difficult to associate with ontology elements. For example, columns that describe numerical property values are not easy to interpret when a column title is abbreviated (e.g. the column *Qty* of table 2.1). Last, in tables that represent experimental results, the list of arguments of a semantic relation may vary depending on the existence of some implicit constants. For example, for a given food product, the amount of calories can be described per 100 grams (implicit argument), or for a specified quantity of the product (an additionnal argument).

Products	**Qty**	**Lipids**	**Calories**
whiting with lemon	100 g	7.8 g	92 kcal
ground crab	150 g	11.25 g	192 kcal
endive	250 g	18.75 g	31 kcal

Figure 2.1 – *Nutritional composition of some food products*

In the e.dot project, our aim was to handle all these difficulties. In addition, we did not want to ask users to correct or validate results that could be proposed by our system. Our challenge was to automatically generate a representation of the tables that can be used by an ontology-based query engine to help users to interpret the data coming from these tables even if the semantic enrichment process has led to an incomplete or erroneous result.

2.2.3 Contributions

We have focused on data tables included in HTML, Excel or Pdf documents. Since XML is a standard for data exchange, that allows to combine text and data with a flexible structure described in DTDs or XML schemas, we have decided to represent the enriched tables in XML. We have defined:

- A method that aims to automatically represent tables extracted from heterogenous Web documents in a XML document. In this XML representation, most of the tags and values are expressed using a controlled vocabulary defined in an ontology. Furthermore, original table elements (cell values, titles), ambiguities or unidentified columns are kept. Semantic relations are proposed even if some arguments are missing. Some information about the mapping process are retained.

- A Document Type Definition named SML (Semantic Markup Language)

that defines the results that can be obtained by this method. This DTD can automatically be generated from the ontology.

2.2.4 An ontology-based approach to enrich tables semantically

In this approach, tables are first represented in XML using purely syntactic tags (Xtab format). Then, tables are represented using terms and relations described in an ontology so that they can be queried using its vocabulary (SML format).

The XTab format

In the settings of the e.dot project, the INA-PG partner has designed tools to extract and represent simple tables in XML, using purely syntactic and domain-independent tags. According to the XTab format they have defined, tables are automatically represented using a list of lines, each line being composed of a list of cells. Besides, when it is possible, titles are extracted. As an illustration, the XTab representation of the table in Figure 2.1 is shown in Figure 2.2.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<table>
<table-title>Nutritional Composition of some food products</table-title>
<column-title>Products</column-title>
<column-title>Qty</column-title>
<column-title>Lipids</column-title>
<column-title>Calories</column-title>
<nb-col>4</nb-col>
<content>
<line>
<cell>whiting with lemon</cell>
<cell>100 g</cell>
<cell>7.8 g</cell>
<cell>92 kcal</cell>
</line>
<line>
<cell>ground crab</cell>
<cell>150 g</cell>
...
</line>
</content></table>
```

Figure 2.2 – *XTab Representation of figure 2.1*

The Sym'Previus ontology

In the setting of a previous project, the Sym'Previus project, the INA-PG partner had developed a terminological resource dedicated to the risk assessment domain. At this time, and for this project, INRA researchers were not interested in representing class instances. Indeed, representing different crab instances or listeria instances was not meaningful for them. This particular resource, erroneoulsy referred to as an ontology, and named the Sym'Previus ontology, is composed of:

1. a term taxonomy which contains 428 terms of the domain (food, microorganism, experimental factors, ...) which are organized by a specialization relation that was noted $\preceq$;

2. 25 semantic relations between terms of the taxonomy. A semantic relation r is characterized by its signature $attrs(r)$ composed of the set of attributes of the relation. The elements of $attrs(r)$ belong to the term taxonomy. For instance, the relation *foodFactorMicroorganism* has the signature *(food, factor, microorganism).*

XTabToSML

XTabToSML allows to enrich XTab documents with tags and values provided by the Sym'Previus ontology. We have defined a representation formalism named SML – *Semantic Markup Language* – where table lines are not represented by cells anymore but by a set of semantic relations between columns. Figure 2.3 shows a part of the SML document which is automatically generated from the first line of the XTab document of Figure 2.1. The main part of the document is inside the ***content*** element. The content is described by a set of lines (***rowRel***) and each line is now a set of semantic relations (like, for example, in Figure 2.3, ***foodLipid or foodCalorie***).

```
<table>
<table-title>Nutritional Composition of some food products </table-title >
<column-title> Product </column-title>
...
<content>
<rowRel additionalAttr="yes">
<foodLipid relType="completeRel">
<food attrType="Normal">
<ontoVal indMap="Jaccard"> whiting fillets </ontoVal>
<ontoVal indMap="Jaccard"> green lemon </ontoVal>
<originalVal> whiting with lemon </originalVal>
</food>
<lipid attrType="Normal">
<ontoVal indMap="notFound"/>
<originalVal> 7.8 g</originalVal>
</lipid>
<attribute attrMatch="notFound" attrType="Generic">
<ontoVal/>
<originalVal> 100 g</originalVal></attribute>
</foodLipid>
<foodCalorie relType="completeRel"> ... </foodCalorie>
<foodAmountLipid relType="partialNull"> ... </foodAmountLipid>
</rowRel>
...
</content> </table>
```

Figure 2.3 – *SML Representation of the nutritional composition of food products*

The SML representation of a relation is composed of the set of attributes that appears in the signature of the relation described in the ontology (e.g. *foodLipid*(*food*, *lipid*)). A set of ontology terms represented inside the XML tag *ontoVal* is associated with each cell value using similarity measures. In the

example, two different ontology terms are proposed for *whiting with lemon*: *green lemon* and *whiting fillets*. The original cell value is kept inside the XML tag *originalVal*.

Indicators that keep track of each step of the mapping process are represented as XML attributes. Thus, the row-level indicator *additionalAttr* is used to know if there exist columns that are not associated to a detected semantic relation. The indicator *reltype* allows to know if all the attributes of a relation have been recognized in the table. The indicator *attrType* enable to store if a column has not been identified or how it has been identified (cell values, constant detected in the title of the table ...). The indicator *indMap* specifies the similarity measure that has been used to map a cell value with an ontology term. These indicators are then used by the query engine to choose how answers must be presented to the user.

The generality of the SML representation is ensured by the automatic generation of the SML DTD from an ontology which contains a taxonomy of terms and relations.

In order to represent the semantic relations expressed in one table using this SML format, we perform two steps:

- The first step consists in identifying terms of the taxonomy which can represent the columns of the table. This allows to obtain the schema of the table.
- The second step consists in discovering candidate semantic relations that can be represented using these mapped columns.

Identification of the columns

The identification of the columns of the data table is based on two kinds of information: the content of the columns and the title used in case the content of the column is not helpful enough.

We first try to associate a term of the taxonomy with each value belonging to the column. Syntactical similarity measures are used in order to find similar terms (lemmatization, inclusion or intersection of sets of words, edit-based measures). When no mapping were found using these syntactical similarity measures, the approach exploits Hearst patterns to discover new subsumption relations on the web with PANKOW (Cimiano et al. (2004)).

Then we search for common generalizers of these terms. We only look for terms of the taxonomy that appear at least one time as an attribute of a relation signature in the ontology. The set of all these *A-terms* is noted AT. The use of a threshold th allows us to associate a generalizer with a given column even if all the values of that column have not been recognized . The set of all A-terms that verifies this constraint is noted $ATCandidate(Col, th)$:

$$ATCandidate(Col, th) = \{t \mid t \ in \ AT \ and \ \frac{|sub(t, Col)|}{|Col|} \geq th\}$$

where *sub(t,Col)* is the set of values of Col that are subsumed by t.

Among these candidates, we select the most specific A-terms that subsume the largest set of values (noted ATRep(col,th)). If no representative A-term

has been found by using this procedure, we exploit the title of the column if available. Finally, if no A-Term has been found, we keep the column in the SML document and we associate the generic A-term named *attribute* with it.

As an illustration, in the table of Figure 2.1, the most specific A-term that subsumes 2/3 of the mapped terms is the A-term *Food*. However, the column Qty has not been identified because it only contains numeric values and the title is an abbreviation. The generic A-Term *attribute* is associated with it. Finally, *lipid* and *calorie* have been associated with the last two columns thanks to the exploitation of their title.

The schema of a table *tab*, noted *tabSch(tab)*, results from this first step. It is the finite set of couples $(col, ATRep(col, th))$ that can be found for a given threshold *th*. The schema of the table *Tab2* shown in Figure 2.1 is:

tabSch(Tab2)={(1,food) (2,attribute) (3,lipid),(4,calorie)}

Identification of the semantic relations

To identify one or several semantic relations represented in a table, we compare this prior discovered schema with the attributes appearing in the signatures of the semantic relations in the domain ontology. In most of the cases, the mappings are not perfect. Some relations are partially or completely represented in the table.

A relation is ***completely represented*** if each attribute of its signature subsumes or is equal to a distinct A-term of the table schema. For example, in table of Figure 2.1, *foodLipid* and *foodCalorie* are completely represented.

A relation is ***partially represented*** if it is not completely represented in the table and if at least two attributes of its signature subsume or are equal to different A-terms of the schema of the table. Some of them are ***partially represented with null attributes*** when an attribute of the semantic relation has not been associated to a column of the table schema. For example, in the table of Figure 2.1, the semantic relation *foodAmountLipid* is partially represented in the table schema *tabSch*, since the attribute *amount* is not represented (reltype="partialNull"). In this example, the generic attribute created for the column *Qty* represents precisely the missing attribute *Amount*. Some of them are ***partially represented with constant values*** when one of the relation attributes corresponds to a constant value which appears in the title of the table. For instance, let {(1,food),(2,factor)} be the table schema computed from the table tab_3 of Figure 2.4. In this table schema, the relation *foodFactorMicroorganism* is partially represented and the missing attribute *Microorganism* is represented by a constant value *Listeria Monocytogenes* which appears in the table title "Doubling time of ***Listeria Monocytogenes*** in foodstuffs". In our approach, this constant is propagated into all the instances of the relation. Figure 2.5 presents the SML representation of the *foodFactorMicroorganism* relation.

Since unidentified data can be useful to interpret the answers provided by the query engine, we also add the set of generic attributes of the table schema to the discovered semantic relations. Actually, one of these additional attributes may be a missing attribute of the relation. Besides, this attribute

Products	Doubling time (h)
Minced meat	30[1]
Cured raw pork	3.6[1]
Frankfurters	9[1]

Figure 2.4 – *Doubling time of Listeria monocytogenes in foodstuffs*

```
<table>
<content>
<rowRel additionalAttr="no">
...
<foodFactorMicroorganism relType ="partialConst">
<food attrType="Normal">...</food>
<factor attrType="Normal"> ... </factor>
<microorganism attrType ="Const"> listeria monocytogenes </microorganism>
</foodFactorMicroorganism>
...
</rowRel> ...
</content> </table>
```

Figure 2.5 – *SML representation of partially represented relations with attributes in constants*

can add a contextual information which may modify the user's interpretation of the relation (e.g. thanks to the generic attribute created for *Qty*, the user can guess that *"250g of chicken corresponds to 312 Kcal"* instead of discovering that *"100 g of chicken corresponds to 312 Kcal"*).

When no relation has been found in the table schema, a generic relation named *relation* is generated in the SML document. In this way, we keep semantic links between values even if this link has not been identified. Thus, it is also possible to query the SML documents by means of lists of (ontology) key-words.

Querying of SML documents

To query SML documents, XQuery queries can rely on the ontology and on the SML DTD. The list of indicators that are represented in the SML document can be exploited during the query evaluation. An indicator can be used to evaluate the risk of a mapping error, to rank the possible answers or to show some of the original table information that can be used to better interpret the results.

Let's consider that a user looks for the quantity of lipid in 100 g of crab. The evaluation of this query consists in searching in the SML document for the subtrees such that the parent node is *foodLipid* and such that there is an element *ontoVal* that contains the value "crab". As the indicator *additionalAttr* has the value "yes", the query engine displays the additional information *150g* that has not been mapped and can show the associated column title. Additionaly, the original value *ground crab* is displayed since *indMap* indicates that the jaccard similarity measure has been used to associate *ground crab* to *crab*. The evaluation of this query performed on the document of Figure 2.3 is presented in Figure 2.6.

<table>
<title> Nutritional composition of some food products </title>
<food> **ground crab**</food> <lipid>**11.25 g**</lipid>
<similarity>jaccard</similarity>
<additionalattr>**150 g**</additionalattr>
<columnTitle> Qty</columnTitle> </table>

Figure 2.6 – *A possible structure of the query answer*

2.2.5 Evaluation Overview

The approach has been tested on the risk assessment domain represented in the Sym'Previus ontology. In this evaluation, we show the capacity of our system to recognize relations of the ontology in the XTab tables. We compared the results provided with our automatic method with a manual enrichment done by an expert. We have collected 50 tables from the Web and we have compared the recall, the precision and the F-measure for the different kinds of semantic relations. This experimentation has shown that when partially identified relations (PR) are kept the recall significatively increases (recall(CR)=0.37 and recall(CR&PR)=0.60) and the precision does not fall much (0.61 to 0.56).

2.2.6 Summary

The XTab2SML approach was developed to represent web tables using a controlled vocabulary and a flexible representation. The flexible format SML has been defined to help the user to interpret the data provided by web tables. In this format, original table values, additional arguments, or incomplete mappings are retained and a list of indicators that keeps track of some information related to the mapping process can be used by the query engine to adapt the proposed answers. In this project, many difficulties were related to the numerical columns that were not so easy to map with the attributes described in the Sym'Previus ontology. Once the project e.dot was finished, the INRA partner has pursued this work by designing a system named ONDINE. ONDINE relies on an ontolexical resource in which simple and complex measure units are represented. This representation is used to improve the identification of semantic relations that involve numerical values. Furthermore, ONDINE allows approximate answers to be retrieved by comparing preferences expressed as fuzzy sets with fuzzy RDF annotations that are generated for web tables (Buche et al. (2013)).

2.3 Semantic annotation of HTML documents

2.3.1 Context

I have worked on the semantic annotation of HTML documents from 2008 to 2013 with Mouhamadou Thiam (PHD student co-advised with Nacéra Bennacer and Chantal Reynaud), Yassine Mrabet (PHD student co-advised with Nacéra Bennacer and Chantal Reynaud), Nacéra Bennacer (Researcher, SUPELEC), Chantal Reynaud (PR, University of Paris Sud) and Moussa Lo (researcher, University Gaston Berger, Senegal). These works take place

within the SHIRI project, funded by DIGITEO Labs, which was a cooperation between SUPELEC, and University Paris Sud (LRI). The aim of SHIRI was to propose ontology-based approaches to semantically annotate heterogeneous HTML documents. The purpose was to allow users to access to relevant parts of documents as answers to their ontology-based queries. SHIRI uses RDF/OWL for representation of resources and SPARQL for querying. The approach that enriches the lexical component of an ontology has been described in the international conference DEXA (Thiam et al. (2009)), the approach SHIRI-Querying that annotates and queries HTML document nodes has been first published in the workshop SEMMA collocated with ESWC (Thiam et al. (2008)), then in the international conference CAISE (Mrabet et al. (2010)). Finally, the approach REISA that discovers property instances in HTML documents has been published in the french journal RIA (Mrabet et al. (2014)), and the international conference WISE (Mrabet et al. (2012)).

2.3.2 Motivation

Many annotation tools have been proposed in the last decade for automatic annotation of documents. These annotations can be used to enrich onto-lexical resources or knowledge bases.

Sometimes the discovered annotations are not accurate (e.g. using *Event* metadata instead of *Conference* metadata) or incomplete. Some approaches (Hurtado et al. (2006), Corby et al. (2006)) deal with semantic imprecision by approximating concepts and relations expressed in user queries using an ontology (e.g. exploiting subsumption, contextual closeness, path of semantic relations). Other works combine ontology-based search and classical keyword search (Castells et al. (2007), Bhagdev et al. (2008)) in order to deal with incomplete annotations. Indeed, the use of keywords may increase the recall by retrieving instances that are not reachable using semantic annotations.

However, even if imprecision and incompleteness can be handled with such approaches, it is sometimes difficult to locate precisely all class instances in a document since some of them may be blended in the free text of the heterogeneous and unstructured documents.

Additionaly, while noticeable advances are achieved for annotation of concept instances, annotation of semantic relations remains a challenging task when the structures and the vocabularies of target documents are heterogeneous or when interphrastic relations are described in documents. In such cases, using lexico-syntactic patterns (Suchanek et al. (2009), Aussenac-Gilles and Jacques (2006), Suchanek et al. (2006)) or regular structures (Limaye et al. (2010), Hignette et al. (2009), Buitelaar and Siegel (2006)) won't allow to extract all the semantic relations that are described in documents. Thus, complementary approaches are needed in order to discover these relations with regularity-independent methods.

The aim of the ontology-based approaches developped in SHIRI is to exploit the syntactic structure of heterogeneous HTML documents in an annotation process. This structure is either used to better rank the annotations when they appear in the most structured parts of the documents or to propose semantic relations that are difficult to discover with patterns.

2.3.3 Contributions

We have defined three different approaches in the setting of the SHIRI project:

- A first approach exploits documents to enrich the lexical component of an onto-lexical resource. More precisely, the aim is to learn new named entities (e.g. *ISWC* can be associated to the concept Conference) and alternative concept labels (e.g. *deep learning* is a term that can be associated to the concept Topic). The originality of the approach is to efficiently combine a mapping process based on usual similarity measures with web invocations (Cimiano et al. (2005)). The web is used as an external resource when named entities or terms are unknown and are not similar to any existing terms of the ontology. To exploit the web, queries containing unmapped terms are submitted to a search engine in order to find candidate labels that can be mapped with the ontology (e.g. if it is found that *"ISWC is an international conference"* on the web, the system will try to map *International conference* with existing terms of the ontology). Using the proposed *Extract-Align* algorithm, the number of Web invocations decreases with the number of processed documents. Indeed, (1) the more the ontology is enriched by new terms, the more a term candidate can be directly mapped and (2) all the terms that cannot be mapped thanks to the web are stored and exploited to prevent the system for invocating the web when these terms belong to other documents.

- An annotation approach that relies on an annotation model called *SHIRI-Annot*. In this model, HTML nodes are annotated by ontology concepts and different types of annotations are distinguished depending on the semantic heterogeneity of the HTML document nodes. Based on the SHIRI-Annot model, an ontology-based search approach called *SHIRI-Querying* has been defined to reformulate user queries and rank the possible reformulations.

- A second annotation approach, named *REISA*, enriches RDF-OWL knowledge bases with property assertions using HTML documents annotated with concept instances. This approach does not use lexico-syntactic patterns but exploits structural closeness between document entities. Ontology axioms and existing facts are used to limit the selection of wrong or redundant property assertions.

In this document, I will only present an overview of SHIRI-Querying and REISA.

2.3.4 SHIRI-Querying: supporting semantic search on more or less structured documents

We have proposed an ontology-based search approach called *SHIRI-Querying*. This approach relies on the annotation model *SHIRI-Annot* (Mrabet et al. (2010)). In this model, the granularity of the annotation is the document node (e.g. HTML tags). This allows bypassing the possible imprecise localisation of instances in texts. Each node is annotated as containing one or several instances of different concepts of a given domain ontology. Moreover, the

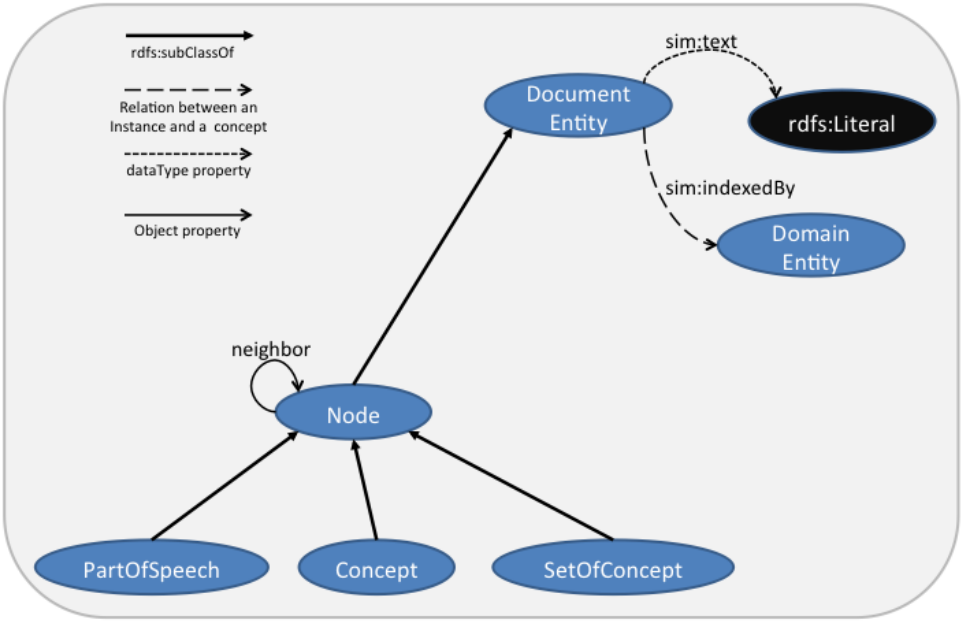


Figure 2.7 – *SHIRI-Annot model*

annotation model also allows representing structural links between document nodes. This enables dealing with incompleteness of semantic relations. These annotations are used by an ontology-based query engine to look for document nodes that are annotated by domain concepts. We propose to reformulate the contraints expressed using domain concepts and relations into constraints expressed on annotated document nodes. In this reformulation step, the document structure and the annotations are used to propose first nodes that are supposed to be the more accurate.

Annotation model and annotation rules of SHIRI-Annot

The SHIRI-Annot annotation model (Figure 2.7) represents the concepts and relations defined for the annotation task. In this model, a *Document entity* represents a part of document that is annotated. In the HTML context, we have considered that these annotated parts are HTML nodes. The literals associated by the *text* attribute are the textual contents of annotated nodes. Nodes are indexed by domain concepts using *isIndexedBy* property. Additionally, the following types of node are defined:

- *Concept* represents document nodes that contain only one instance of a domain concept.

- *SetOfConcept* represents document nodes that contain several instances of comparable concepts (with regard to the subsumption relation).

- *PartOfSpeech* represents document nodes that contain several instances of different domain concepts.

- The *neighbor* relation expresses that two annotated document nodes are neigbours in the HTML document tree.

In Figure 2.9, I show an extract of two HTML documents where HTML nodes are typed differently. In this example, the first HTML node of Document 1 that contains *31 may - 4 june 2009, Heraklion, Greece* is a $PartOfSpeech$ since its text refers to domain concepts that are not comparable (i.e. several dates and several locations). This node is indexed by the domain concepts *Date* and *Location*. The HTML node that contains *12th international Conference on Datawarehousing and Knowledge Discovery* is a *Concept* node indexed by *Event* since it contains only one instance of *Event*. Finally, the node which contains the text *Bilbao, Spain* is a $SetOfConcept$ since its text refers to several instances of *Location*. Since the neighbor relations will be used to replace a property in the user Query, the approach only represents *neighor* relations between nodes that contain concept instances that may be linked by a domain property. This is checked using the domain and range properties.

Instances of Shiri-Annot metadata are generated using a set of rules (see table 2.8).

Notation	Sens
$singleTerm(n)$	only one term or named entity has been found in a HTML node.
$containInstanceOf(n, C_o)$	The node contains one instance of the domain concept C_o.
$dist(n_1, n_2)$	distance between HTML nodes that contains n_1 et n_2 in the HTML document tree.
$comparable(n)$	All the domain concepts that are used to index the node n are comparables vs the subsumption relation.
μ	Maximum distance defined to consider that two nodes are neighbours in the HTML document.

Condition	Generated annotation
$singleTerm(n) \wedge containInstanceOf(n, C_o)$	$\rightarrow type(n, Concept) \wedge indexedBy(n, C_o)$
$\neg singleTerm(n) \wedge containInstanceOf(n, C_o) \wedge comparable(n)$	$\rightarrow type(n, SetOfConcept) \wedge indexedBy(n, C_o)$
$containInstanceOf(n, C_o) \wedge \neg singleTerm(n) \wedge \neg comparable(n)$	$\rightarrow type(n, PartOfSpeech) \wedge indexedBy(n, C_o)$
$indexedBy(n, C_o) \wedge indexedBy(n', C'_o) \wedge \exists R$ tq. $domain(R, C_o) \wedge range(R, C'_o) \wedge dist(n, n') < \mu$	$\rightarrow neighbor(n, n')$

Figure 2.8 – *Rules used to annotate document nodes in $SHIRI - Annot$*

Query Reformulations

In Figure 2.9, it is also shown that an ontology-based user query *Qo* needs to be reformulated to obtain a set of document nodes. The reformulation of a query Q_0 is a query q_i obtained by a first *neighborhood-based* reformulation followed by a composition of elementary *Concept*, *PartOfSpeech*, and *SetOfConcept* reformulations. We have ordered the set of reformulated queries, to obtain first the queries that are the most relevant when the structure of the HTML document is considered. The most relevant queries involve few document nodes (i.e. all the searched information is located in few distinct parts of the document) and involve nodes that are structured (i.e. they do not contain many different kinds of information).

We defined a **model-based query** q as a quadruplet (P, S, F, D) where :

- P is a basic graph pattern which complies with a model (i.e. an ontology $\mathcal{O}$ or the annotation model $\mathcal{A}$). $V(P)$ denotes the set of variables of P.

-F is a constraint defined by a logical combination of boolean-valued expressions.

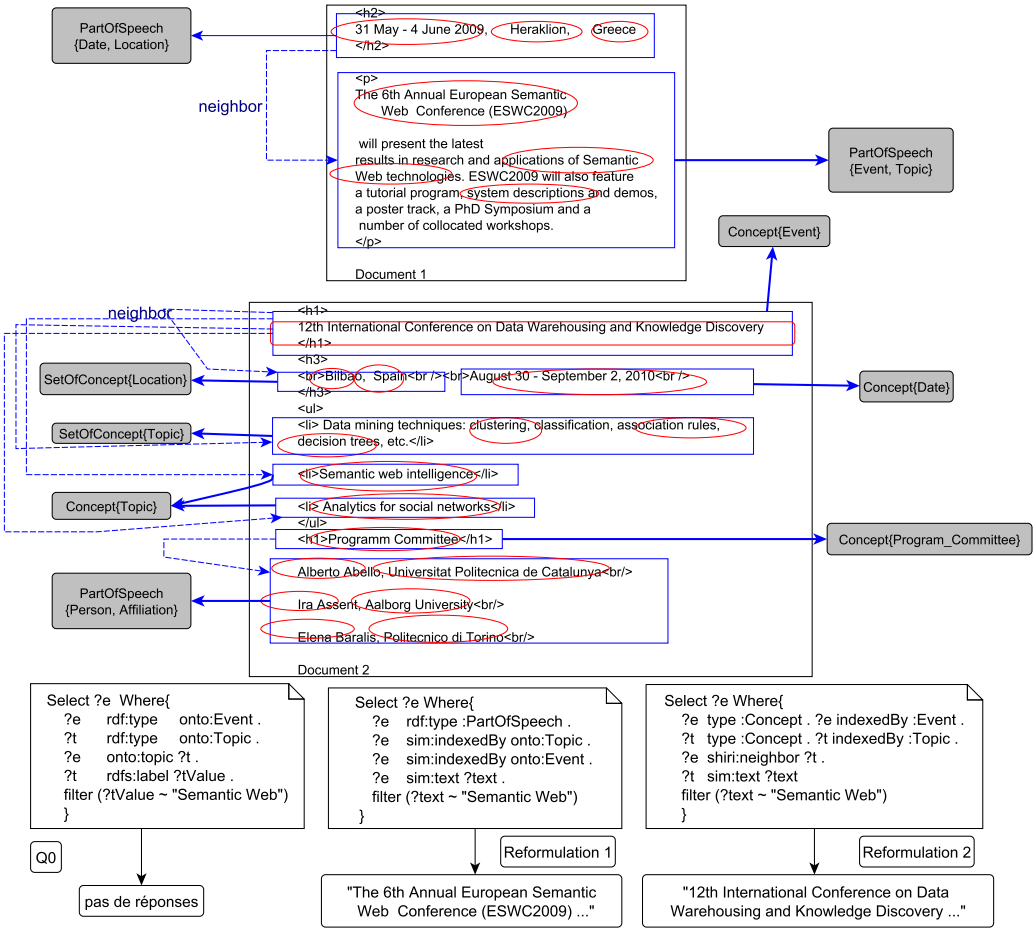


Figure 2.9 – *Extracts of Html documents where nodes are annotated differently*

-S is the set of variables that appear in the $SELECT$ clause of the query.
-D is an $\mathcal{A}$-compliant RDF dataset to be matched with P and F.

Example: An $\mathcal{O}$-based query q_0 is defined by (P_0,F_0,S_0,D) where :
P_0 ={ ($?e$, rdf:type, onto:Event), ($?t$, rdf:type, onto:Topic),($?e$,onto:topic,$?t$)
($?t$, rdfs:label,$?text$) }
F_0 : {$?text = "semanticWeb"$} and S_0 : { $?event$, $?topic$, $?text$ }

Figure 2.10 shows different elementary reformulations that can apply to this ontology-based user query that looks for scientific events related to a *semantic web* topic.

Neighborhood-based Reformulation: The aim of this first neighborhood-based reformulation is to look for annotated document nodes (instead of class instances). It exploits the structural neighborhood of document nodes in order to find nodes that may be related by the semantic relations expressed in the user query (i.e. the ontological relation of a given triple is substituted by a $neighborOf$ relation).

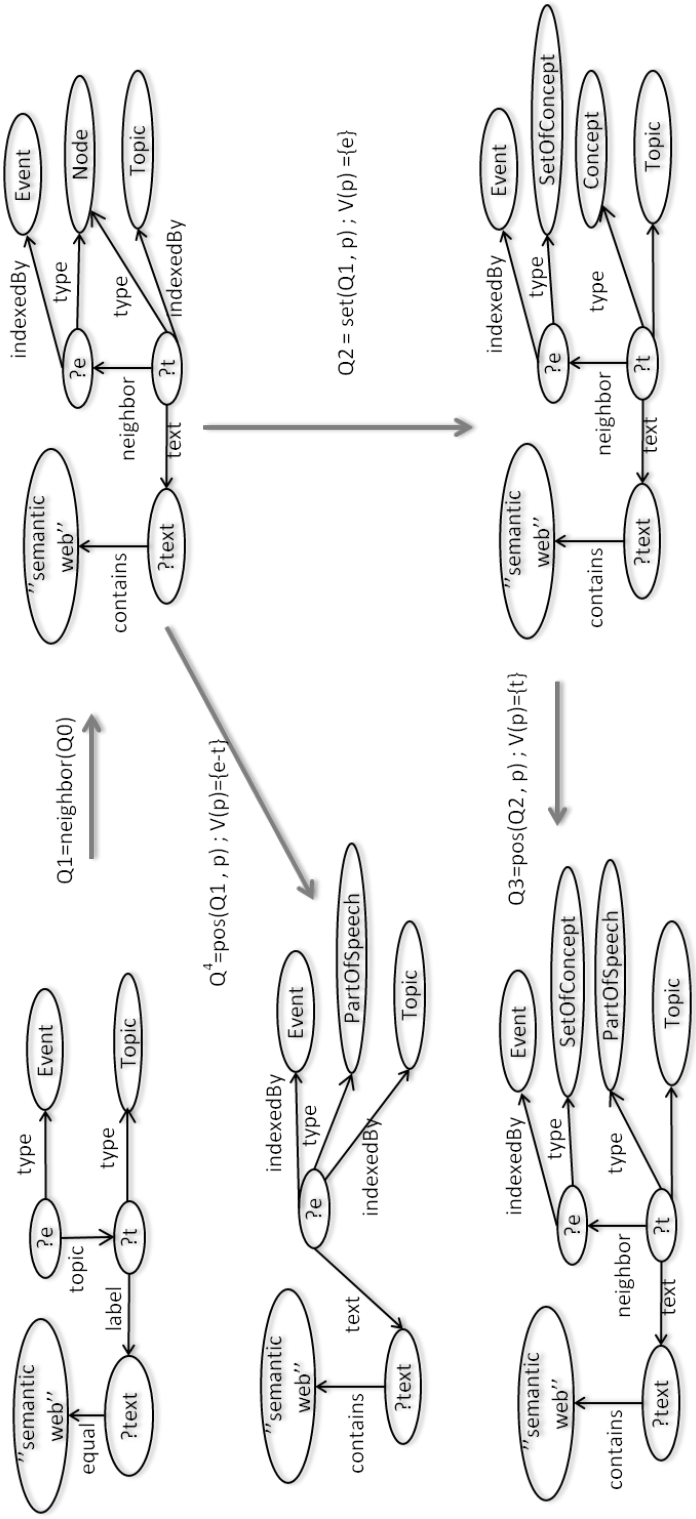


Figure 2.10 – *Examples of elementary reformulations in SHIRI-Querying*

Concept Reformulation This reformulation just adds the triple $node_i$ rdf:type shiri:Concept in the query for an existing $node_i$ of the query.

***PartOfSpeech* Reformulation:** The $PartOfSpeech$ reformulation denoted f_{pos} assumes that the required semantic relations can be found between instances aggregated in the same node. This reformulation, when applied to a query Q_1, transforms a subgraph g_{cn} of P_1 into g_s:

- $\forall v_i \in g_{cp}$ that is a $Node$, if $< v_i$ sim:indexedBy $C_i >\in g_{cp}$ then $< node_s$ sim:indexedBy $C_i >\in g_s$
- $\forall v_i \in g_{cp}$ that is a $Node$, if $< v_i$ sim:text $text_i >\in g_{cp} \wedge \exists e_i(text_i) \in F_1$ then $< node_s$ sim:text $text_s >\in g_s \wedge e_i(text_s) \in F_1$

***SetOfConcept* Reformulation:** Similarly, the $SetOfConcept$ reformulation denoted f_{set}, when applied to Q_1 transforms a subgraph (with comparable concepts) g_{cp} of P_1 into g_s:

- $\forall v_i \in g_{cp}$ that is a $Node$, si $< v_i$ sim:indexedBy $C_i >\in g_{cp}$ then $< node_s$ sim:indexedBy $C_i >\in g_s$
- $\forall v_i \in g_{cp}$ that is a $Node$, si $< v_i$ sim:text $text_i >\in g_{cp} \wedge \exists e_i(text_i) \in F_1$ then $< node_s$ sim:text $text_s >\in g_s \wedge e_i(text_s) \in F_1$

Reformulations Construction Plan: The reformulation of a query $q_0(P_0, F_0, S_0, D)$ is a query $q_i(P_i, F_i, S_i, D)$ obtained by the composition of the first *neighborhood-based* reformulation followed by *Concept*, *PartOfSpeech*, and *SetOfConcept* reformulations that are applied to connected subgraphs of the reformulated query. To rank the queries and prefer the queries that involve few nodes that are as structured as possible, we have defined the following order:

Definition 1. Let $N(q)$, $Pos(q)$ and $Sets(q)$ be resp. the number of $neighborOf$, $PartOfSpeech$ and $SetOfConcept$ nodes in a query q. The order $\preccurlyeq$ is defined s.t. $q_i \preccurlyeq q_j \leftrightarrow ((N(q_i) > N(q_j)) \vee ((N(q_i) = N(q_j)) \wedge ((Pos(q_i) > Pos(q_j)) \vee ((Pos(q_i) = Pos(q_j)) \wedge (Sets(q_i) \geq Sets(q_j))))$

We have proposed an algorithm named DREQ (Dynamic Reformulation and Execution of Queries Algorithm) that allows constructing and executing the reformulated queries with respect to $\preccurlyeq$. When $DREQ$ is stopped at a given order, the answers are those retrieved by the best constructed queries.

Evaluation overview

SHIRI-Querying has been implemented and experimented to study how the precision and the recall measures vary according to the order relation. The $neighborOf$ relation is defined as an undirected path of length d in the HTML /XML tree. We also study how d influences the results. The two experimented datasets belong to the scientific conferences and publications domain.

The first dataset is composed of annotated publication references provided by the DBLP XML repository, the INRIA HTML server and the HAL XML repository. It consists of more than 10.000 RDF triples describing 1000

publications. We submitted a set of queries looking for conferences, their dates, locations, papers and authors. A precision of 100% and a recall of 100% were reached with an order threshold of 9 and $d \leq 3$. A higher distance d leads to almost 0% precision (in two data sources, each paper is associated to all conferences). The 100% values for the recall and the precision measures are due to the regular structure of each data source. However, each data source has a different and specific structure and the $DREQ$ reformulations were able to integrate answers from all sources.

The second corpus consists of RDF annotations of 32 call-for-papers web sites and is consequently very heterogeneous. These annotations (consisting of 30.000 RDF triples) were generated automatically using $SHIRI - Extract$ (Thiam et al. (2009)). We then submitted a set of 15 queries. Without reformulation, all queries have no answers (0% recall), while we obtained a 56% recall and 72% precision by using the $DREQ$ algorithm for $d \leq 7$. The results show that domain relations can often be retrieved between instances located in close document nodes. The precision variations have shown that the order relation was relevant to rank the answers.

2.3.5 REISA: a controlled enrichment of knowledge bases with HTML documents

Annotation Problem considered in REISA

The approach REISA considers an annotation context where the following elements are provided:

- a set of HTML documents,
- one or several populated ontologies (i.e. RDF/OWL knowledge bases) that have eventually been aligned,
- annotation tools that can discover named entities in the free text of HTML documents and that can associate them to class instances that are described in the considered knowledge bases (desambiguisation), when they exist.

The aim of REISA is to enrich knowledge bases with property instances using the set of HTML documents annotated with concept instances. To discover property instances this approach do not use lexico-syntactic patterns but exploits the structural closeness between document entities. We exploit the following hypothesis: annotated document entities which are closer to each other in the documents are more likely to be semantically related. To limit the generation of wrong or redundant property assertions, ontology axioms and existing facts are used to filter the obtained results.

Semantic Integration Model

To represent in a homogeneous way the knowledge bases and the results obtained by the annotation tools, we have defined a simple model called *Semantic Integration Model* (SIM) (see Figure 2.11). Domain entities (i.e. concept

and relation instances), document parts and annotation links between both elements can be represented using this model.

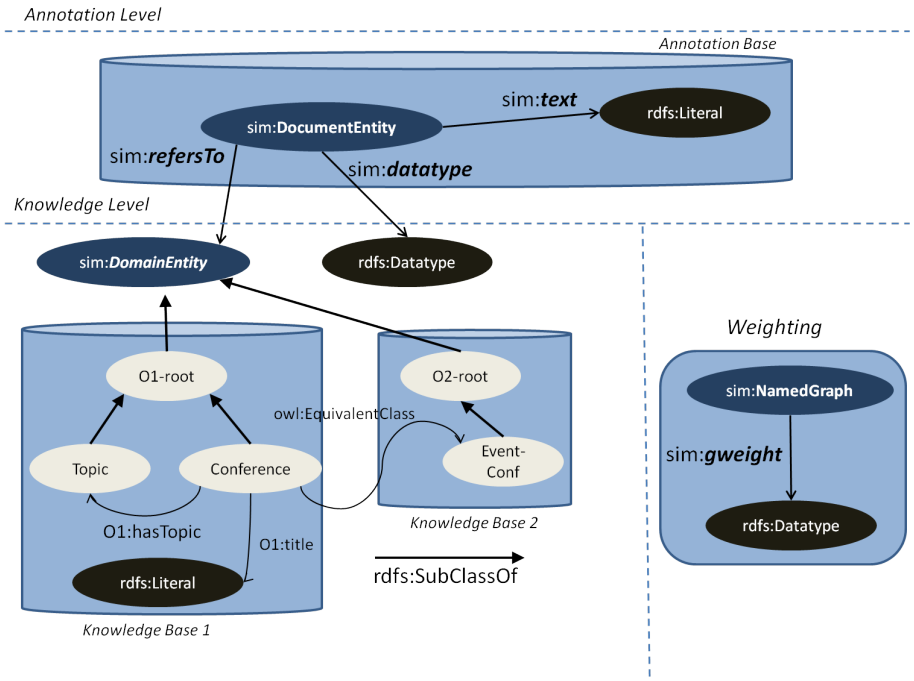


Figure 2.11 – *Semantic Integration Model (SIM)*

A *knowledge base KB* can be defined by the couple (O, T_O) where O is an OWL ontology defined by the quadruplet (C_O, R_O, A_O, X_O). C_O is the set of concepts, R_O is the set of properties defined between concepts, A_O is the set of properties defined between concepts and literals. X_O is the set of axioms which define, for example, properties' domains and ranges, (inverse) functionality of the defined properties, equivalence links generated by ontology alignment processes. T_O is the set of facts that describes the instances of the ontology's concepts and properties.

The *Annotation Base AB* is a set of document entities and annotation links generated by annotation tools between the document entities and the knowledge bases. In our approach, document entities may be (i) document nodes defined in the DOM tree of the document (in that case, their textual content *sim*:*text* is the concatenation of all their child text nodes) or (ii) a text window which may correspond to a named entity or to word sequences. Annotated document entities are linked to concept instances of the knowledge bases by the *sim*:*refersTo* property. The links between document entities and datatypes (*rdfs*:*datatype*) are represented by the *sim*:*datatype* property.

In this approach, a domain expert can assign a confidence value of 1 to validated knowledge bases and inferior weight values to knowledge bases or annotation bases constructed automatically by annotation tools. The facts that are produced by our enrichment approach are weighted automatically using the closeness of the annotated entities in the document. We have exploited RDF named graphs in order to associate a confidence measure to fact sets from the knowledge bases. Named graphs are RDF graphs identified

by a URI[1]. The weight of a named graph represents the weight of the RDF triples it contains. The property *sim:gweight* allows to represent this weight as a real number between 0 and 1.

Figure 2.12 shows an extract of a HTML document, an extract of KIM knowledge base (WKB) and an extract of the annotation base generated by the KIM platform and expressed using the model SIM. The confidence value assigned to these annotations is 0.9.

Extract of a HTML document
... **Laos** traces its history to the kingdom of Lan Xang ... took over **Vientiane** with ... along the Annamite mountains in **Vietnam**. ... tools discovered in northern **Laos** attest ... communities along the **Mekong** River ...   ... Following the military defeat of Japan ... the Viet Minh occupied **Hanoi** and proclaimed a provisional government, which asserted national independence ... ...
Extract from the WKB (KIM) knowledge base
@prefix graphs: <http://lri.fr/reisa/graphs/> graphs:knowledgebase = { kimkb:Laos.0 **rdf**:**type** onto:Country kimkb:Laos.0 **onto**:**partOf** kimkb:Continent.2 kimkb:Continent.2 **rdf**:**type** onto:Continent kimkb:Continent.2 **rdfs**:**label** "Asia" kimkb:Vientiane.0 **rdf**:**type** onto:City kimkb:Vientiane.0 **onto**:**capital** kimkb:Laos.0 }
Extract from the Annotation base of the document

graphs:annotationsbase = {
corpus:doco/html/body/div/p[3]/a.0 **rdf**:**type** sim:DocumentEntity
corpus:doco/html/body/div/p[3]/a.0 **sim**:**refersTo** kimkb:Vietnam.0
corpus:doco/html/body/div/p[3]/a.0 **sim**:**text** "Vietnam"
corpus:doco/html/body/div/p[3].12 **sim**:**refersTo** kimkb:Laos.0
corpus:doco/html/body/div/p[3].12 **sim**:**text** "Laos"
corpus:doco/html/body/div/p[3].20 **sim**:**refersTo** kimkb:Mekong.0
corpus:doco/html/body/div/p[3].20 **sim**:**text** "Mékong"
corpus:doco/html/body/div/p[3]/a[2].0 **sim**:**refersTo** kimkb:Hanoi.0
corpus:doco/html/body/div/p[3]/a[2].0 **sim**:**text** "Hanoi"
}
graphs:knowledgebase sim:gweight 1
graphs:annotationsbase sim:gweight 0.9

Figure 2.12 – *Extracts from the WKB knowledge base, a HTML document and its associated annotation base*

The enrichment module of REISA

REISA consists in three main modules (cf. figure 2.13):

- the integration module which allows to represent the *KBs* and the *AB* according to the SIM model,
- the enrichment module which uses the *KBs*, the *AB* and the documents structure in order to produce new property instances and
- the interrogation module which allows to answer user queries from the *KBs* and ranks the answers according to the weights of returned facts.

[1] http://www.w3.org/2004/03/trix/

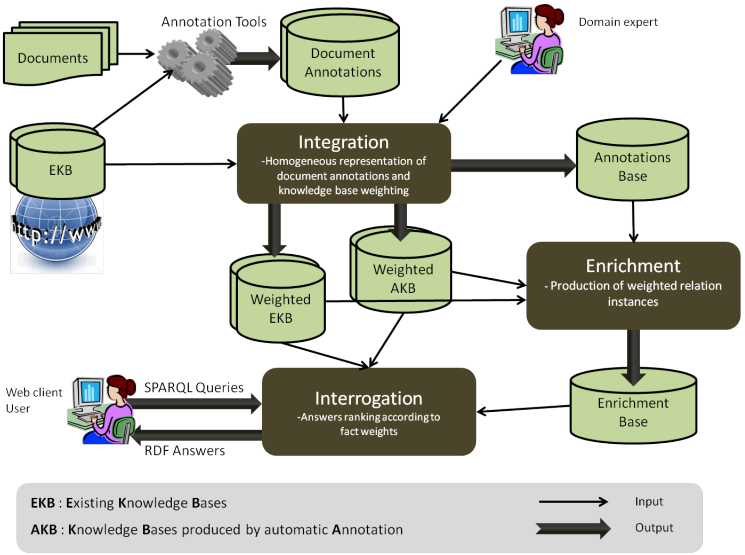


Figure 2.13 – *REISA modules*

In the enrichment module, we look first for document entity pairs that refer to class instances or literals such that these class instances or literals are compatible with a property domain and range. These document entities are called *semantic neighbors*. They are used to construct candidate property instances. These candidate properties are weighted by a confidence measure computed according to the distance between the annotated document entities. In a second step, the domain ontology and the already available property instances in the *KBs* are used to filter these candidates.

Semantic Neighbors. Two concept instances i_1 and i_2 are said to be semantic neighbors at a distance d for a property P, noted $V_P^d(i_1, i_2)$, if two document entities e_1 and e_2 referring to i_1 and i_2 are at a distance d from each other in the document, and the instances i_1 and i_2 belong to concepts that are respectively domain and range of P :
The distance d between two document entities is the shortest distance between the deepest document nodes containing them in the (cleaned) DOM tree of the document.
$refersTo(e_1, i_1) \wedge refersTo(e_2, i_2) \wedge distance(e_1, e_2) = d \wedge$
$type(i_1, C_1) \wedge type(i_2, C_2) \wedge domain(P, C_1) \wedge range(P, C_2) \rightarrow V_P^d(i_1, i_2)$

In a similar manner, a concept instance i and the textual content l of a document entity are semantic neighbors at a distance d for a property P if the litteral associated to this text and a concept of the domain instance are respectively range and domain of the property P.

The enrichment process consists in constructing the weighted property instances and their corresponding neighborhood named graphs until a distance threshold μ is reached. A Neighborhood named graph, noted $G^{w(d)}$, contains the property instances produced by enrichment from annotated document entities at a distance d in the document. The weight of a neighborhood named graph, $w(d)$, is defined such that it is inversely proportional to the distance d. It is computed as follows, for $\alpha \leq 1$ a constant:

$$w(d) = \frac{\alpha}{d+1} \tag{2.1}$$

The value of α is fixed by the domain expert who associated weight values to the existing knowledge bases. This allows (i) to keep some coherence between all the weight values and (ii) to put a maximum limit to the weight of the candidate relations at distance 0 so that it is inferior to he weight of existing (valid) knowledge. A property instance $P(i_1,i_2)$, between two concept instances i_1 and i_2, is added to the graph $G^{w(d)}$ if:

- V_P^d (i_1,i_2)
  i_1 and i_2 are semantic neighbors at distance d for the relation P.

- $\neg\exists$ P$(i_1,i_2) \in G^p$ s.t. $p > w(d)$
  The fact does not already exist in graphs having a better weight.

- if P is functional $\neg\exists\ z$ s.t. $P(i_1, z) \in G^1$
  If the property is declared as functional in the ontology, and i_1 is already linked to an instance z for the property P in a valid KB, then one of these two facts holds: (1) i_2 and z refer to the same world entity (should be linked with *owl*:*sameAs*) and, accordingly, $P(i_1,i_2)$ does not bring new knowledge or (2) $P(i_1,i_2)$ is a wrong fact.

- if P is inverse functional then $\neg\exists\ z$ s.t. $P(z, i_2) \in G^1$
  Similar reasoning.

Similar enrichment constraints are defined for datatype properties.

Illustrating example

Figure 2.14 shows an example of neighborhood graphs generated by the enrichment module according to the extracts presented in figure 2.12. Here, the weights were computed with α=0.9 and a distance threshold μ=2 (structural distances 0, 1 and 2).

```
@prefix graphs: <http://lri.fr/reisa/graphs/>
graphs:candidates.distance.0 {
kimkb:Mekong.0 dbpedia:country kimkb:Laos.0
}
graphs:candidates.distance.1 {
kimkb:Mekong.0 dbpedia:country kimkb:Vietnam.0
}
graphs:candidates.distance.2 {
kimkb:Hanoi.0 kim:capital kimkb:Vietnam.0
}
graphs:candidates.distance.0 sim:gweight 0.9
graphs:candidates.distance.1 sim:gweight 0.45
graphs:candidates.distance.2 sim:gweight 0.3
```

Figure 2.14 – *Example of three neighboring graphs constructed by enrichment*

In this example, REISA generated three candidate property instances. We can see that the property *kim*:*capital* is asserted between “Hanoi” and “Vietnam”

which are at distance 2 in the document but not between "Vientiane" and "Vietnam" which are at distance 0. This is due to the fact that the *kim:capital* property is functional, which allowed to avoid proposing the city "Vientiane" as capital of "Vietnam", since the city is already known as the capital of another country (*Laos*) in the knowledge base (see knowledge base extract in figure 2.12).
The interrogation module of REISA can be used to query the enriched knowledge bases based on the ontologies. Moreover, this module performs a query rewriting to use the named graphs weights in the answers ranking. For example, figure 2.15 shows a query rewriting that uses the average function to aggregate the weights of the returned facts (other aggregation measures could be used).

User query	Rewrited query

SELECT ?c ?t WHERE {
?c rdf:type kim:Country
?t rdf:type kim:City
?t kim:capital ?c
}
	SELECT ?c ?t WHERE {
	GRAPH ?g1 { ?c rdf:type kim:Country }
	GRAPH ?g2 { ?c rdf:type kim:City }
	GRAPH ?g3 { ?t kim:capital ?c }
	?g1 sim:gweight ?p1
	?g2 sim:gweight ?p2
	?g3 sim:gweight ?p3
	} ORDER BY avg(?p1,?p2,?p3)

Figure 2.15 – *Example of SPARQL query rewriting*

2.3.6 Evaluation Overview

The aim of the experiments was to evaluate the precision of the facts produced by our enrichment method according to (i) their weight (or neighboring distance) and (ii) the application of the filtering step that exploits the KB axioms and the KB facts.

In a first experimentation, we have constructed a knowledge base (KB) by combining extracts from the DBLP-RDF knowledge base [2] which describes bibliographic references (5608 instances of *swrc:Proceedings* and their descriptions) and extracts from the WKB knowledge base of KIM[3] related to cities (3093 instances), countries (502 instances) and locations (12484 instances). Some of the locations of the conference events were missing in DBLP-RDF. So, we have annotated 511 HTML files from call-for-paper websites with the KIM platform. The aim was to exploit these annotations to enrich the KB with conference locations that are generated with REISA. The correctness of the retrieved locations were verified manually in the HTML pages describing the events.
The experiments have shown that the quality of the new facts varies according to the neighborhood distance. Precision decreases significantly when the neighboring distance increases. At a distance 0, almost one fact on two is correct for the *hasCity* relation and 76.5% of the facts are correct for the *hasCountry* relation.

The results also show that the functionality-based control can have an important impact on the facts precision. For example, for the property *hasCountry*, this control increases the precision from 58,8% to 76,5% at distance 0.

[2] http://thedatahub.org/dataset/fu-berlin-dblp
[3] http://www.ontotext.com/kim/semantic-annotation

In a second experimentation in the geographic domain, we used Wikipedia articles and DBpedia as a reference knowledge base. We constructed the KB automatically with a SPARQL query submitted to DBpedia and considered all the 27 documents corresponding to mountain ranges of France. In a similar approach to (Gerber and Ngomo (2011)), we used Wikipedia surface forms in order to annotate the documents with concept instances. A surface form is the text of hyperlinks to other Wikipedia documents. As each wikipedia page is associated to one concept instance from DBpedia, a *sim*:*refersTo* link is asserted between the surface form and the DBpedia instance associated to the document pointed by the hyperlink.

We have focused on the relations *mouthCountry* and *sourceCountry* which indicate the mouth and source countries of a river and *BodyOfWater/Country* which indicates the country of a body of water. A total of 429 relation instances was produced by REISA for these three relations.

Like in the first experiment, the precision of the discovered relation instances decreased when the distance threshold increased. The overall precision of all relations is 71,42% at distance 2 (no relation instances were produced with distances 0 and 1) and decreased progressively to 49,54% at distance 10. This second experimentation has also shown that the functionality-based control allows to significantly increase the precision.

2.3.7 Summary

This approach is complementary to "classical" approaches that uses lexico-syntactic patterns. Using REISA, weighted property instances are discovered in documents annotated with concept instances without requiring any lexico-syntactic or structural regularity. The experiment results have shown the interest of exploiting knowledge bases in controlling the enrichment process. However, obviously, these semantic controls are only possible if the used annotation tools is able to discover reference links between document entities and existing instances described in the knowledge bases. Additionally, if the domain ontology defines many possible relations between the same pair of concepts, this approach is expected to be less effective. The aim of this work was to propose a complementary knowledge acquisition strategy that can be combined with other approaches. The REISA approach could be exploited to populate (semi-)automatically RDF knowledge bases by validating candidate facts with domain experts when a high recall is needed. Note that, the validated facts could then be used to enhance the automatic controls performed by REISA.

2.4 Conclusion

In this chapter, all the presented approaches aim to semantically annotate documents and in all these approaches the syntactic structures of the web documents are taken into account.

The XTab2SML approach was developped to annotate web tables in a flexible manner. In this approach the table structure is exploited to discover implicit semantic relations between data related to the risk assessment domain.

In the setting of the SHIRI-project, we have considered more or less structured HTML documents. The first proposition was to distinguish document nodes that are rather structured from nodes that contain a lot of information. In this approach the structure is also used to represent neigbor links that express a structural closeness between annotated nodes. To exploit these annotations, ontology-based queries are reformulated using DREC. DREC constructs the queries that will obtain the closed and structured document nodes first. In this first context, we have not considered a populated ontology, so it was not possible to use existing instances to improve the quality of the obtained results. In the second approach (REISA), we aim to enrich a OWL/RDF knowledge base with property instances that are discovered in HTML documents. In this second approach, the structure is used to propose property instances that involve concept instances that have been located in neighbor nodes of the HTML document. Since this strategy may lead to many incorrect property instances, we have used the ontology axioms and the already described instances to filter the obtained results.

We are conscious that text-driven knowledge acquisition tools cannot be considered as stand-alone approaches when they are used to enrich onto-lexical resource or to populate knowledge bases. Either they are complemented by human expert interventions to integrate the acquired knowledge in the existing ontology, either a high-quality knowledge base is available but incomplete and the discovered knowledge is kept separated, enriched with confidence degrees and provenance information.

Data linking

3

3.1 Introduction

In this chapter, I present the works done on and around the data linking topic.

The data linking problem consists in deciding whether some kind of semantic links holds between two object descriptions. In particular, identity links can be declared when two object descriptions refer to the same real world entity (e.g. the same person, the same company).

Let us consider two RDF descriptions of the painting *"Femmes d'Alger dans leur appartement"* (Eugène Delacroix). The first one is provided by BNF institution (see Figure 3.1) while the second by DBpedia (see Figure 3.2).

The data linking problem is to decide that these two descriptions refer to the same painting.

Figure 3.1 – *Extract of the RDF description of the painting Woman in Algier in the BNF dataset (HTML viewer)*

Identity links can be used to bring together information from different sources in order to create a new, richer dataset. Indeed, based on these links, it is possible to combine information about the same real-world entity prior stored in several repositories.

Figure 3.2 – *Extract of the RDF description of the painting Woman in Algier in the DBpedia dataset (HTML viewer)*

"Linking data about people from multiple sources can tell a bigger story than analysing data from just one source. For example, comparing the data on women who received the HPV vaccination with data on women who developed cervical cancer showed that overall, the vaccination was effective in reducing cervical cancer" [1].

The problem of data linking was introduced by the geneticist Newcombe (Newcombe et al. (1959)) and was first formalized by Fellegi and Sunter (Fellegi and Sunter (1969)). Since then, various approaches have been proposed in different areas and under different names. In relational databases, record linkage, also known as data cleaning, duplicate detection or object matching, is a classical problem (see Winkler (2006a), Köpcke and Rahm (2010), Elmagarmid et al. (2007) for a survey). The goal of record linkage is to identify records corresponding to the same entity from one or more datasets.

In the Semantic Web, one can refer to this problem as Data Interlinking or Reference Reconciliation where a reference is a URI of an object. In this context, data descriptions are coming from different sources and these sources can be heterogeneous, built in an autonomous way by different users or organisations. Even if some schemas such as Dublin Core or FOAF are more and more exploited, datasets often use their own schemas and this schema heterogeneity is a major cause of data description mismatches between sources. Extensive research work has been done to align ontologies through class or property mappings (see Shvaiko and Euzenat (2013) for a survey). For example, the class *Work* of the BNF dataset and the class *Oeuvre* of the DBpedia dataset can be mapped by an equivalence relation.

The conformity to a same global ontology does not prevent variations between data descriptions. First, different conventions and vocabularies can be used to describe data. Data descriptions can contain syntactic variations in the literal values, for example a person name can be *"J. Doe"* or *"Doe, John"*, a date can be represented using various formats, a country name can be expressed in different languages. Many works have shown that there is no

[1] Australian Institute of Health and Welfare website : http://www.aihw.gov.au/data-linking/

universally best similarity measure to compare literal values (Sarawagi and Kirpal (2004), Cohen et al. (2003)). Second, information can be incomplete, i.e., the values of some properties can be missing. Third, descriptions may also contain erroneous values. More generally, as data descriptions can be created and updated independently in different sources, this can lead to different descriptions representing the same real world entity.

Furthermore, scalability needs to be taken into account when designing data linking approaches: approaches that compare every pair of objects will not succeed to scale when data are numerous. Therefore, strategies are needed to reduce the search space.

Another important data linking issue is to be able to provide explanations about data linking results. This issue is particularly challenging when data linking is based on numerous steps that involve complex numerical functions and optimization processes.

The main contributions I present in this chapter is first LN2R, a Logical and Numerical approach for Reference Reconciliation. Then, I introduce KD2R and SAKey, two approaches that aim to discover composite keys from RDF datasets. Finally, I present an outgoing work that aims to discover logically existing invalid identity links.

3.2 LN2R: A Logical and Numerical approach for Reference Reconciliation

3.2.1 Context

I have worked on data linking approaches from 2005 to 2013 with Fatiha Saïs (PhD-Student co-advised with Marie-Christine Rousset, then MCF), Marie-Christine Rousset (PR), Souhir Gahbiche (Master Student co-advised with Fatiha Saïs). These works were initially motivated by the data integration task defined in the industrial project PICSEL 3 (Production d'Interfaces à base de Connaissances pour des Services En Ligne), a project realized in collaboration with France Telecom R&D. The logical approach L2R has been first described in the proceedings of the national french conference Extraction et Gestion des Connaissances (EGC) in 2007, then in the proceedings of the twenty-second conference on Artificial Intelligence (AAAI) in 2007 (Saïs et al. (2007)). The combination of logical and numerical approaches performed in LN2R has been described in the Journal On Data Semantics (JODS), 2009 (Saïs et al. (2009)). We have participated to the OAEI Instance Matching track the results of which have been published in the Workshop Ontology Matching collocated with the International Semantic Web Conference (ISWC) in 2010 (Saïs et al. (2010)). Finally, some other results linked to these two data linking tools have been proposed: an approach that aims to explain linking results (Gahbiche et al. (2010b;a)), an approach that can be applied to improve the scalability of the linking process (Pernelle and Saïs (2012)), and finally, a linking approach that can be applied when property mappings are partially known (Pernelle et al. (2013a)).

3.2.2 Motivation

Nowadays, the RDF datasets are generally large, making it impossible to define identity links manually. Thus, many approaches have been proposed to discover identity links (semi-)automatically (see (Ferrara et al. (2011), Elmagarmid et al. (2007), Winkler (2006b) for a survey)).

Existing approaches can be distinguished according to various criteria. An approach can be considered as local (instance based) or global (graph-based). A local approach exploits the common properties of two resource descriptions to decide if an identity link can be generated (Volz et al. (2009), Nikolov et al. (2012a)). Some of the local approaches exploit the unstructured description of the text appearing in the properties without distinguishing which value corresponds to which property (Cohen (2000), Bilke and Naumann (2005)). This kind of approach is useful, either to discover fast candidate pairs for reconciliation or to link data when property mappings are not known. Global approaches exploit properties to propagate reconciliation decisions or similarities between pairs of resources (Dong et al. (2005), Bilgic et al. (2006)). These global approaches take into account a larger set of information.

An approach can also be considered as supervised or knowledge-based. A supervised approach exploits training data (a set of existing links) to learn rules or functions that can be used in the linking process. For instance, in (Dong et al. (2005), knowledge about property discriminability can be learnt on training data. A knowledge-based approach exploits either expert knowledge or ontology semantics to define linking rules. For example, Silk (Volz et al. (2009)) or RiMOM (Li et al. (2009)) are based on human provided linking rules. Nikolov et al. (2012a) exploit some ontology semantics to generate links (functional properties). An approach such as Object Coref is both ontology-based and supervised since the ontology is used to discover a subset of the identity links and then, these links are exploited to learn the most discriminative set of properties for one given resource (Hu et al. (2011)).

Furthermore, approaches are applicable in different contexts. In some approaches, the data should conform to the same (eventually aligned) ontology while some others can be applied even when the schema is not known. Moreover, there exist approaches that discover ontology mappings and identity links in the same process (Suchanek et al. (2011)).

Most of the existing methods infer only reconciliation decisions. They do not aim to produce non reconciliations. However, to reduce the size of the reconciliation space, some methods apply a candidate selection step to select object pairs that are likely to be linked. Thus, they implicitly consider a set of non-reconciliation decisions. This is the case for the so-called blocking methods introduced in (Newcombe and Kennedy (1962)) and used in approaches such as (Baxter et al. (2003), Song and Heflin (2011)).

In 2005, when we have begun to work on this subject, as far as we know, only one data linking approach was existing for RDF data that conform to RDFS or OWL ontologies (Dong et al. (2005)). However, this approach does not exploit the ontology semantics to construct linkage rules. In the setting of the project PiCSEL3, we have studied the problem of data linking (i.e. reference reconciliation) when data sources are described relatively to one rich ontology. The aim was to define a global and unsupervised approach that was guided by the knowledge described in the ontology.

3.2.3 Contributions

We have proposed a knowledge-based, unsupervised approach for data linking, based on two methods that can be applied separately or in combination:

- a logic-based method called L2R. This method infers both reconciliation and non-reconciliation decisions.
- a numerical method called N2R. This method computes similarity scores for each pair of references.

In both methods the (non) reconciliation decisions or the similarity scores are propagated to other reference pairs. Since we exploit the axioms that are declared in the ontology and some constraints that can be declared on the data sources, the methods are not so sensitive to domain and data changes. The approach is based on Semantic Web standards: RDF, OWL-DL and SWRL.

3.2.4 L2R: a Logical approach for Reference Reconciliation

Reference reconciliation problem considered in L2R : In this work, we consider that the descriptions of data coming from different sources conform to the same ontology (possibly after an ontology alignment step).

Let *Reconcile*[2] be a binary predicate. *Reconcile*(X, Y) means that the two references denoted by X and Y refer to the same world entity.

The reference reconciliation problem considered in L2R consists in extracting from the set $I_1 \times I_2$ of reference pairs two subsets REC and NREC such that :

$$\begin{cases} REC = \{(i,i') |\ Reconcile(i,i')\} \\ NREC = \{(i,i') |\ \neg Reconcile(i,i')\} \end{cases}$$

Set of reconciliation rules

The distinguishing features of L2R are that it is global and logic-based: some of the ontology axioms and some constraints that are declared on the considered datasets are automatically translated into logical rules that express dependencies between reconciliations. These Horn rules enable to infer reconciliations and non-reconciliations among a subset of data item pairs, and synonymies between literal values. The advantage of such a logical approach is that it guarantees correct results, under the assumption that the schema and data are error-free.

The ontology axioms that are considered are the following : disjunctions between classes, functional properties, inverse functional properties and composite keys. I illustrate the rules that are automatically generated from the ontology using the example of the functional properties.
For each object property R declared as functional, the following rule $R6.1(R)$ is generated :

$$R6.1(R) : Reconcile(x,y) \wedge R(x,z) \wedge R(y,w) \Rightarrow Reconcile(z,w)$$

[2]Reconcile and not Reconcile can also be expressed in OWL by using *sameAs* and *differentFrom* predicates.

For example, if an object property *Located* which relates references of museums to references of cities is declared functional, the reconciliation of two museums will lead to reconcile the two cities where they are located in.
For each datatype property A declared as functional, the following rule $R6.2(A)$ is generated by L2R:

$$R6.2(A) : Reconcile(x, y) \wedge A(x, z) \wedge A(y, w) \Rightarrow SynVals(z, w)$$

This rule allows inferring synonymies (*SynVals*) between literal values.

We have also proposed to generate rules from constraints that can be declared on the data sources. I illustrate these rules by the case of the *UNA*.

The UNA states that two data items of the same data source having distinct references refer to two different real world entities (and thus can never be reconciled). We have introduced the unary predicates Src1 and Src2 in order to label each reference according to its original source ($Srci(x)$ means that the reference x is coming from the source Si). Thus, if the UNA assumption is stated on the source $S1$, it is translated automatically by the two following rules :

$$R1 : Src1(x) \wedge Src1(y) \wedge (x \neq y) \Rightarrow \neg Reconcile(x, y)$$

$$R3 : Src1(x) \wedge Src1(z) \wedge Src2(y) \wedge Reconcile(x, y) \wedge (x \neq y) \Rightarrow \neg Reconcile(z, y)$$

$R1$ expresses that two distinct references coming from the same source cannot be reconciled. $R3$ means that one reference coming from a source $S2$ can be reconciled with at most one reference coming from a source $S1$.

Sometimes, the UNA cannot be stated but a Local UNA (called *LUNA*) can be declared for some properties. For example, if the LUNA is declared for the object property $Authored$, relating paper references to people references: it expresses that there are no duplicates in the set of authors of a given paper. The LUNA assumption is also translated automatically into rules.

In total, there are 17 rule patterns that can be exploited to generate rules in L2R, using either ontology axioms, constraints declared on the data sources or properties of the Reconcile predicate (e.g. transitivity).

Reasoning method

In order to infer reconciliations and non-reconciliations, we have used an automatic reasoning method based on the resolution principle Robinson (1965). This method applies to the clausal form of the set of rules $\mathcal{R}$ and a set of facts $\mathcal{F}$ describing the data: the RDF facts corresponding to the data description in the two sources S_1 and S_2, facts of the form $Src1(i)$ and $Src2(j)$ for each reference $i \in I_1$ and each reference $j \in I_2$, synonymy facts of the form $SynVals(v_1, v_2)$ for each pair (v_1, v_2) of literal values that are identical (up to some punctuation or known syntactic variations), non synonymy facts of the form $\neg SynVals(v_1, v_2)$ for each pair (v_1, v_2) of distinct basic values of a functional datatype property. For instance, $\neg SynVals(''2014'', ''2015'')$, $\neg SynVals(''France'', ''Australia'')$ can be easily added.

The reasoning aims at inferring all unit facts in the form of $Reconcile(i, j)$, $\neg Reconcile(i, j)$, $SynVals(v1, v2)$ and $\neg SynVals(v1, v2)$.

Several resolution strategies have been proposed so that the number of computed resolutions to obtain the theorem proof are reduced (for more

details see Chang and Lee (1997)). We have chosen to use the *unit resolution* (Henschen and Wos (1974)), a resolution strategy where at least one of the two clauses involved in the resolution is a unit clause, i.e., reduced to a single literal. Indeed, the unit resolution is complete for refutation in the case of Horn clauses without functions (Henschen and Wos (1974)). Furthermore, the unit resolution method is linear with respect to the size of clause set (Forbus and de Kleer (1993)).

The unit resolution algorithm that we have implemented consists in computing the set of unit instantiated clauses contained in $\mathcal{F}$ or inferred by the unit resolution on $\mathcal{R} \cup \mathcal{F}$. Its termination is guaranteed because there are no function symbols in $\mathcal{R} \cup \mathcal{F}$. Its completeness for deriving all the facts that are logically entailed has been stated.

Note that the instantiated predicates *Synvals* and $\neg$*SynVals* are saved in dictionaries that can be exploited to improve the results of new applications of the numerical or the logical tool.

Illustrative example

I now illustrate the reconciliation and non-reconciliation decisions that can be inferred by L2R on a simple example of data and schema.

Source S1 :
museumName(S1_m1,*"LE LOUVRE"*); contains(S1_m1,S1_p1);
located(S1_m1,S1_c1); cityName(S1_c1,"Paris");
paintingName(S1_p1,*"La Joconde"*);museumAddress(S1_m1,*"99, r. RIVOLI"*);

Source S2 :
museumName(S2_m1,*"musee du LOUVRE"*); located(S2_m1,S2_c1);
contains(S2_m1,S2_p1); contains(S2_m1, S2_p2); cityName(S2_c1, *"Ville de Paris"*);
paintingName(S2_p1, *"Abricotiers en fleurs"*); paintingName(S2_p2, *"Joconde'*);
museumAddress(S2_m1,*"rue Rivoli"*);

Figure 3.3 – *Example of RDF data of cultural places domain.*

The UNA is stated in both sources. These data conform to the simple ontology presented in Figure 3.4. In this ontology, the properties *Located*, *PaintedBy* and *MuseumName* are declared as functional properties while the properties *PaintingName* and *Contains* as inverse functional.

The successive application of the unit resolution on the knowledge base $\mathcal{R} \cup \mathcal{F}$, presented in the Figure 3.3, allows inferring the set of (non) reconciliations and synonymy facts presented in Figure 3.5.

L2R: evaluation overview

In order to evaluate the quality of the results of a reference reconciliation method, well-known measures defined in the Information Retrieval (IR) domain can be employed. In this context, *Precision*, *Recall* and *F-Measure* can be defined as follows :

- **Precision** is the ratio of correct reconciliations and non-reconciliations among those found by the method.

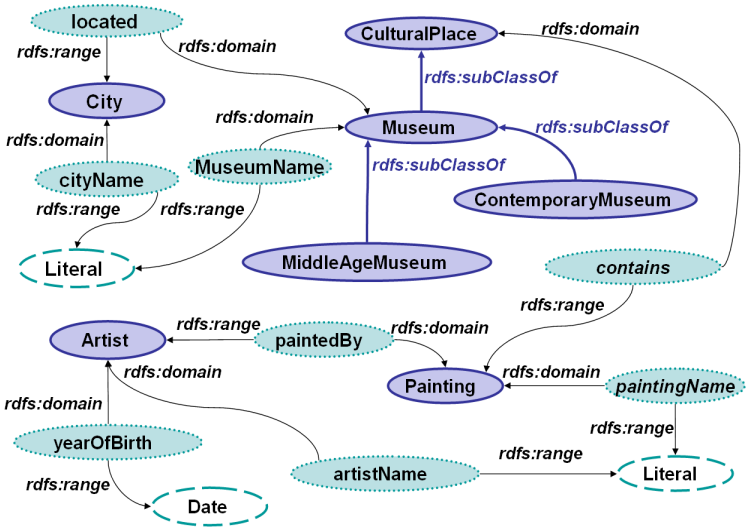


Figure 3.4 – *Extract of a simple ontology about cultural places*

Knowledge base

$\mathcal{R}$= { $R1 : \neg\ Src1(x) \vee \neg\ Src1(y) \vee \neg Reconcile(x,y)$ /UNA
$R2 : \neg\ Src2(x) \vee \neg\ Src2(y) \vee \neg Reconcile(x,y)$ /UNA...
$R5(Painting, City) : \neg\ Painting(x) \vee \neg\ City(y) \vee \neg Reconcile(x,y)$/Disjunction
$R5(Painting, Museum) : \neg\ Painting(x) \vee \neg\ Museum(y) \vee \neg Reconcile(x,y)$ /Disjunction
$R5(City, Museum) : \neg\ City(x) \vee \neg\ Museum(y) \vee \neg Reconcile(x,y)$ /Disjunction ...
$R6.1(Located) : \neg\ Reconcile(x,y) \vee \neg\ Located(x,z) \vee \neg\ Located(y,w) \vee Reconcile(z,w)$ /Functional prop.
$R6.2(MuseumName) : \neg\ Reconcile(x,y) \vee \neg\ MuseumName(x,z)$
$\vee \neg\ MuseumName(y,w) \vee SynVals(z,w)$ /Functional prop.
$R6.2(CityName) : \neg\ Reconcile(x,y) \vee \neg\ CityName(x,z) \vee \neg\ CityName(y,w) \vee SynVals(z,w)$ /Functional prop. ...
$R7.2(PaintingName) : \neg SynVals(x,y) \vee \neg\ PaintingName(z,x) \vee$
$\neg\ PaintingName(w,y) \vee Reconcile(z,w)$ /Inverse Functional prop.
$R7.1(Contains) : \neg Reconcile(x,y) \vee \neg\ Contains(z,x) \vee \neg\ Contains(w,y) \vee Reconcile(z,w)$ /Inverse Functional prop. ...}
$\mathcal{F}$= { $MuseumName(S1\_m1, \text{“}LE\ LOUVRE\text{”}); Contains(S1\_m1, S1\_p1);$
$Contains(S2\_m1, S2\_p2); CityName(S2\_c1, \text{“}Ville\ de\ paris\text{”}); ...$
$Src1(S1\_m1); Src1(S1\_p1); Src1(S1\_c1); Src2(S2\_m1); Src2(S2\_p1); Src2(S2\_c1)$
$SynVals(\text{“}La\ Joconde\text{”}, \text{“}Joconde\text{”})$}

Reference reconciliation result

SatUnit$(\mathcal{R} \cup \mathcal{F})$ = {...;
$\neg Reconcile(S1\_m1, S1\_c2)$; $\neg Reconcile(S1\_m1, S1\_p1)$; $\neg Reconcile(S1\_p1, S1\_c1)$;
$\neg Reconcile(S2\_m1, S2\_p1)$; $\neg Reconcile(S2\_m1, S2\_c1)$; $\neg Reconcile(S2\_c1, S2\_p1)$;
$\neg Reconcile(S1\_m1, S2\_p1)$; $\neg Reconcile(S1\_m1, S2\_c1)$; $\neg Reconcile(S1\_p1, S2\_c1)$;
$\neg Reconcile(S1\_c1, S2\_p1)$;
$Reconcile(S2\_p1, S1\_p1)$; $Reconcile(S1\_m1, S2\_m1)$; $Reconcile(S1\_c1, S2\_c1)$;
$SynVals(\text{“}musee\ du\ LOUVRE\text{”}, \text{“}LE\ LOUVRE\text{”})$; $SynVals(\text{“}ville\ de\ Paris\text{”}, \text{“}Paris\text{”})$}

Figure 3.5 – *Illustrative example of unit resolution-based reference reconciliation*

- **Recall** is the ratio of correct reconciliations and non-reconciliations found by the method among the whole expected set of correct reconciliations and non-reconciliations.
- **F-Measure** is computed to balance the recall and precision values: $F\text{-}Measure = (2 * Recall * Precision) \div (Recall + Precision)$.

L2R has been evaluated on datasets related to two different domains: scientific publications (CORA dataset) and Hotel descriptions provided by France Telecom R&D. The first dataset is a collection of 1295 citations of 112 different research papers in computer science. This dataset contains duplicates (data cleaning scenario). The second dataset contains 28934 references of hotels located in Europe. These descriptions were distributed in a set of seven data sources which led to a pairwise data integration problem of 21 pairs of data sources. The UNA was stated for each source. Since this method is based on logical inferences, it has a 100% precision, under the assumption that the ontology axioms, the data descriptions and the (not)synvals between literal values introduced are error-free. The aim of the evaluation was to observe the recall. The question was: what is the proportion of the results that can be found by such a logical approach. For both datasets, we were able to find more than 50% of the (non) reconciliation decisions. We have also shown that the recall can significantly increase if the ontology is enriched by new axioms or when (not)synvals are injected. This result is not so impressive when compared to existing numerical approaches. However, in an application, if a precision of 100% is needed, the reconciled pairs obtained by this tool do not require to be checked by an expert. In the same way, if a recall of 100% is needed, the expert does no have to check if there exist reconciliations in set of the non reconciled pairs. Furthermore, this method can be used to filter the reference pairs that are compared using a numerical tool.

Obviously, the size of these two datasets was rather small if we compare it to the size of the datasets available nowadays on the LOD. At that time, such datasets were not common. However, I think that, such a global approach could deal with large datasets if efficient filtering processes are applied to select clusters of data items to be compared.

3.2.5 N2R: a Numerical approach for Reference Reconciliation

In order to complement the partial results of L2R, we have designed a Numerical method for Reference Reconciliation (N2R). This method exploits the results of L2R and allows to compute similarity scores for each pair of references.

Reference reconciliation problem considered in N2R

The reference reconciliation problem considered in N2R consists in, given a similarity function $Sim_r : I_1 \times I_2 \longrightarrow [0..1]$, and a threshold T_{rec} (a real value in $[0..1]$ given by an expert, fixed experimentally or learned on a labeled data sample), computing the following set:

$$REC_{n2r} = \{(i, i') \in (I_1 \times I_2) \backslash (REC \cup NREC) |\ Sim_r(i, i') > T_{rec}\}$$

A similarity function based on the ontology axioms

To distinguish the different impacts of the properties on the similarity of the instance pairs, N2R exploits the semantics of (inverse) functional properties or composite keys that are declared in the ontology. For example, let us consider an object property *isInCountry* that links a city to the country it belongs to. This property is functional: a city is located in one country. Thus,

a strong similarity of two city instances can be propagated to the country instances. We can say that a pair of countries is functionally dependent of a pair of cities. However, a strong similarity of two countries should have a weak impact on the similarity of two cities that are located in these countries. Indeed, the property *isInCountry* is not inverse functional.

In N2R, the mutual influences between similarity scores x_i of a pair of references are expressed in an equation system. Each equation $x_i = f_i(X)$ is of the form:

$$f_i(X) = \max(f_{i-df}(X), f_{i-ndf}(X))$$

The function $f_{i-df}(X)$ is an aggregation function of the similarity scores of the value pairs and the reference pairs of datatype and object properties with which the i-th reference pair is functionally dependent. The function $f_{i-ndf}(X)$ allows to aggregate the similarity scores of the values pairs (and sets) and the reference pairs (and sets) of datatype and object properties with which the i-th reference pair is not functionally dependent.

In this second method, existing similarity measures are used to evaluate the similarity of the literal values (e.g., String, Date, Numerical values) (Cohen et al. (2003)). In the equation system, these similarities are considered as constants.

Some properties are multi-valued (e.g. a person can have a set of phone numbers). In order to compute the similarity between two sets of values we need to take into account their size and also the similarity scores of the pairs of values formed from these two sets. We have introduced a new similarity measure named *SoftJaccard* which is inspired from the *SoftTFIDF* (Cohen et al. (2003)) and the *Jaccard* measures.

Finally, $f_{i-df}(X)$ is defined by the maximum of similarity scores of the value pairs and reference pairs of properties with which the two references ref and ref' are functionally dependent.

$f_{i-df}(X)$ is defined as follows :

$$f_{i-df}(X) = \max\Big(\bigcup_{j=0}^{j=|FD_A(<ref,ref'>)|}(b_{ij-df}), avg\Big(\bigcup_{j=0}^{j=|FD_A^M(<ref,ref'>)|}(b_{ij-dfm})\Big),$$

$$\bigcup_{j=0}^{j=|FD_R(<ref,ref'>)|}(x_{ij-df}), avg\Big(\bigcup_{j=0}^{j=|FD_R^M(<ref,ref'>)|}(x_{ij-dfm})\Big)\Big)$$

The maximum function allows propagating the similarity scores of the values and the references having a strong impact. Indeed, we have considered that, for these properties, the higher similarity value should be propagated: if two mobile phone numbers are the same, even if two contact mails are different, the two people will be considered as highly similar. The similarity scores of the values and the references of a composite key are first aggregated by an *average* function. Thus, if the set of the properties *name*, *first-name* and *birth-date* is declared as a composite key, the similarity scores of the values involved in these three datatype properties will simply be averaged before the selection of the maximum similarity value of all considered keys.

$f_{i-ndf}(X)$ is defined by a weighted average of the similarity scores of the values and the references of properties with which the two references ref and ref' are not functionally dependent. $f_{i-ndf}(X)$ is defined as follows :

$$f_{i-ndf}(X) = \sum_{j=0}^{j=|NFD_A(<ref,ref'>)|} (\lambda_{ij} * b_{ij-ndf}) + \sum_{j=0}^{j=|NFD^*_A(<ref,ref'>)|} (\lambda_{ij} * BS_{ij-ndf}) +$$

$$\sum_{j=0}^{j=|NFD_R(<ref,ref'>)|} (\lambda_{ij} * x_{ij-ndf}) + \sum_{j=0}^{j=|NFD^*_R(<ref,ref'>)|} (\lambda_{ij} * XS_{ij-ndf})$$

λ_{ij} represents the weight of the j-th datatype or object property in the similarity computation of the i-th reference pair. Since we have neither expert knowledge nor training data, λ_{ij} is computed in function of the number of the common datatype and object properties.

Due to the maximum function and the similarity computed for multivalued object properties, the equations are never linear. In order to solve the equation system, we have used an iterative method inspired from the *Jacobi* method (Golub and Loan (1989)) for which we have proved the convergence. At each iteration, the method computes the variable values by using those computed in the precedent iteration. The computation stops when a fix-point with precision ϵ is reached.

Illustrative example

Let us suppose that L2R has been applied, resulting on the non-reconciliations of all the pairs of references coming from the same source and those belonging to two disjoint classes. The only remaining pairs of references to consider for N2R are then:

$< S1\_m1, S2\_m1 >$, $< S1\_c1, S2\_c1 >$, $< S1\_p1, S2\_p1 >$ and $< S1\_p1, S2\_p2 >$.

The similarity score $Sim_r(ref, ref')$ between the references ref and ref' of each of those pairs is modeled by a variable:

- x_1 models $Sim_r(S1\_m1, S2\_m1)$,
- x_2 models $Sim_r(S1\_p1, S2\_p1)$,
- x_3 models $Sim_r(S1\_p1, S2\_p2)$,
- x_4 models $Sim_r(S1\_c1, S2\_c1)$.

The similarity computation can be illustrated on the equation system obtained from the data descriptions of Figure3.3 (without taking into account the property *museumAddress*). The equations, the constants (similarity scores of pairs of literal values), the variables and the weights are given in the Table 3.1. In this example, ϵ is fixed to 0.005.

Variables	Constants	Weights
$x_1 = Sim_r(S1\_m1, S2\_m1)$	$b_{11} = Sim_v(\text{“}LOUVRE\text{”}, \text{“}Musee\ du\ LOUVRE\text{”}) = 0.68$	$\lambda_{11} = \frac{1}{4}$
$x_2 = Sim_r(S1\_p1, S2\_p1)$	$b_{21} = Sim_v(\text{“}La\ Joconde\text{”}, \text{“}Abricotiers\ en\ fleurs\text{”}) = 0.1$	$\lambda_{21} = \frac{1}{2}$
$x_3 = Sim_r(S1\_p1, S2\_p2)$	$b_{31} = Sim_v(\text{“}La\ Joconde\text{”}, \text{“}Joconde\text{”}) = 0.9$	$\lambda_{31} = \frac{1}{2}$
$x_4 = Sim_r(S1\_c1, S2\_c1)$	$b_{41} = Sim_v(\text{“}Paris\text{”}, \text{“}Ville\ de\ Paris\text{”}) = 0.42$	$\lambda_{41} = \frac{1}{2}$

Table 3.1 – *The variables, the constants and the weights of the equation system*

Iterations	0	1	2	3	4
$x_1 = \max(0.68, x_2, x_3, \frac{1}{4} * x_4)$	0	0.68	0.9	0.9	0.9
$x_2 = \max(0.1, \frac{1}{2} * x_1)$	0	0.1	0.34	0.45	0.45
$x_3 = \max(0.9, \frac{1}{2} * x_1)$	0	0.9	0.9	0.9	0.9
$x_4 = \max(0.42, x_1)$	0	0.42	0.68	0.9	0.9

Table 3.2 – *Illustrative example – Iterative similarity computation.*

The equations and the different iterations of the resulting similarity computation are provided in Table 3.2.

The solution of the equation system is $X = (0.9, 0.45, 0.9, 0.9)$. The fixpoint has been reached after four iterations: the computed error vector is then equal to 0.

If we fix a reconciliation threshold T_{rec} at 0.80, then we obtain three reconciliation decisions.

N2R: Evaluation overview

N2R has been implemented and tested on a scientific publication provided by the CORA dataset and another dataset describing hotels provided by France Telecom R&D. The Ontology Alignment Evaluation Initiative (OAEI) provides benchmark datasets for evaluating instance matching approaches. Thus, in 2010, we have also participated to the OAEI'10 Instance matching track *Persons* and *Restaurants*. We were ranked second on these tracks (Saïs et al. (2010)). For $T_{rec} = 1$, both N2R and L2R provide the same results. These experiments have showed that the method is able to obtain a F-Measure which is similar to some supervised methods that were introduced at this period (Parag and Pedro (2004), Dong et al. (2005), Cohen and Richman (2002), without using any labeled data.

The experiments have also showed that the exploitation of the non-reconciliation, inferred by L2R, allows an important reduction of the reconciliation space handled in N2R. For the Cora dataset the size of the reconciliation space is about 37 million of reference pairs. It has been reduced to 32.8% thanks to the no reconciliations inferred by L2R. This reduction corresponds to 12 million of reference pairs. Moreover, the reconciliations inferred by L2R are not recomputed in N2R.

3.2.6 An explanation tool based on Coloured Petri nets

Automatic data linking methods may rise decision errors. Hence, we thought that systems such that N2R need explanation modules to enhance the user confidence in the integrated data and to detect anomalies. In order to explain the similarity scores and the reconciliation decisions, we have proposed an explanation model based on Coloured Petri Nets (CP-Net) which provides graphical explanations to an expert user.

Representation of the N2R equation system using a CP-Net

A classical Petri Net without colours is a directed bipartite graph in which the nodes represent transitions (i.e., discrete events), or places (i.e., states). Directed arcs describe which places are *pre* or *postconditions* for which transitions. Places may hold tokens. A transition is enabled when the number of tokens in each of its input places is at least equal to the arc weight[3] going from the place to the transition. When the transition is fired, the tokens in the input places are moved to the output places, according to arc weights and place capacities. This leads to a new *marking* of the net, i.e., a new state description of all places. In a CP-Net (Jensen (1997)), each token is equipped with an attached data value called the token colour. The data value may be of atomic value type or of complex type (e.g. a record such that the first field is an integer, the second one is a string).

In our approach, a CP-net is exploited to obtain a graphical representation of how a similarity score is computed and propagated by N2R. We have defined how the equation system $F(X) = X$, with $X = (x_1, \ldots, x_n)$, can be translated into a CP-Net called CP_{rec}. Intuitively, the idea is that we create a place for each variable x_i and for each constant c_i. Similarity scores of variables and constants are represented using a token colour (percentages). Each N2R equation is represented by a transition. Dependencies between similarity scores are modeled in the CP-Net by arcs that connect input places to transitions and transitions to output places. In order to show to the user the evolution of the similarity score of a reference pair, old tokens are stored in their corresponding places. The iterative computation of the similarity scores is simulated by successive markings of the CP-Net.

Transformation of F(X)=X into a CP_{rec}: a simple example

The two following equations belong to the equation system presented for the example used in the preceding section:

- $x_1 = max(c_1, x_2, x_3, x_4/4)$, with $c_1 = 0.68$.
- $x_2 = max(c_2, x_1/2)$, with $c_2 = 0.1$.

To obtain the corresponding CP_{rec} shown in Figure 3.6(a), the explanation module automatically creates:

- six places P_{x1}, P_{x2}, P_{x3}, P_{x4}, P_{c1} and P_{c2}. Initially, their token colours are $x_1 = 0$, $x_2 = 0$, $x_3 = 0$, $x_4 = 0$, $c_1 = 10$ and $c_2 = 68$.
- a transition $T1$ which takes the token colours that come from P_{x2}, P_{x3}, P_{x4} and P_{c1} and provides a token for the place P_{x1}. When this transition is fired, $f(x_1)$ is computed.
- a transition $T2$ which takes the token colours that come from P_{x1} and P_{c2}, and provides a token for the place P_{x2}.

[3] A positive integer value that represents the token number that are allowed to cross the transition.

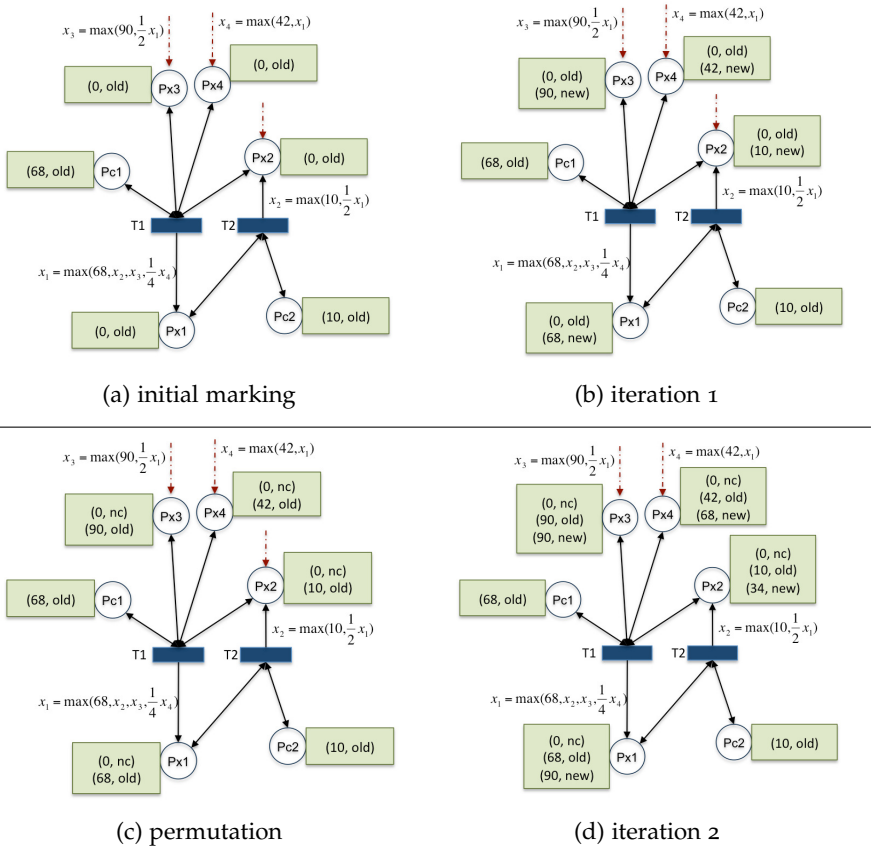


Figure 3.6 – *Simulation of the N2R computation of similarity scores in a Colored Petri Net (Extract)*

The CP-Net of the Figure 3.6 (a) shows the initial marking where a token of colour $(0, old)$ is assigned to each place representing a variable, and a token of colour $(sim_v(v_1, v_2) * 100, old)$ is assigned to each place representing a constant. For example, the token that is assigned to the place P_{c1} is of colour $(68, old)$ (68 corresponds to the similarity score of ("*Musee du Louvre*", "*Le Louvre*")). Firing the transition T_1 allows the computation of the value of x_1. The first iteration (see Figure 3.6.(b)) is terminated when all the transitions of T_e are fired. The similarity scores of the basic values are propagated to the places representing the variables P_{xi} thanks to the creation of a new token with a state equals to *new* and having as value, the similarity score obtained by the evaluation of the similarity function expressed by $Post(t_i, p_{xi})$. After that, a permutation (see Figure 3.6.(c)) is applied on the token colours of the places P_{xi}: the state *old* of the tokens becomes *nc* and the state *new* of the other tokens becomes *old*. During the second iteration (see Figure 3.6.(d)), the values that are computed at the previous (1st) iteration are propagated by creating in each place a new token with a state at *new*. The iteration number after which the fix-point is reached is specified by a parameter which corresponds to the N2R number of iterations.

CP-Net enrichment

We have enriched this model by additional knowledge: (i) schema knowledge about reference descriptions (i.e., classes, properties and literals), (ii) ontological knowledge that semantically explain the dependencies and the impacts in a similarity score computation: (inverse) functional properties, composite keys. To distinguish strong and weak dependencies in the CP-Net, input arcs are coloured differently and to help the user understand this knowledge, we have proposed to associate to these input arcs a textual description which expresses the semantics of the (inverse) functionality. For example, the text "*A museum is located in one and only one city*" can be shown next to the functional property *Located*.

The developed explanation model justifies similarity scores that can be obtained by a global approach such as N2R. Used in an interactive environment, it can be an interesting tool to help expert users accept or revise the discovered links or to help users revise the knowledge that is used in the linking process. Another advantage of such a model is that it can also be used as a simulation tool in order to test the consequences of some changes in the system.

3.2.7 Summary

LN2R is an unsupervised data linking approach that combines a logical step (L2R) and a numerical step (N2R). L2R exploits the ontology axioms (composite keys, functional properties, disjunctions) to logically infer (non) reconciliation decisions. Reconciliation and non reconciliation decisions can then be used as input for the numerical step N2R. N2R exploits ontology axioms to construct an equation system that models dependencies between similarities of class instances. Both steps capture the intuition that if two instance pairs refer to the same world entity then some of their related instances also refer to the same world entity.

To enhance an expert user confidence in the integrated data and to detect anomalies, a graphical explanation model based on Colored Petri-net has been proposed.

3.3 Key discovery for data linking

3.3.1 Context

I have been working on key discovery approaches with Danai Symeonidou (PhD-Student) and Fatiha Saïs (PhD co-supervisor) since 2011. The approach KD2R has been first described in the proceedings of the workshop SWWS in 2011 (Symeonidou et al. (2011)), then in the national conference BDA and in the Journal of Web Semantics in 2013 (Pernelle et al. (2013b)). A demo has been shown in the workshop WOD (Pernelle et al. (2013c)). The second approach called SAKey has been published in the international conference ISWC in 2014 (Symeonidou et al. (2014)). An extension of this approach called C-Sakey has been published in the national conference Ingenierie des connaissances in 2015 (Pernelle et al. (2015)).

These two approaches have been developed in the setting of the ANR project QUALINCA. This ANR research project aims to quantify and improve

the quality level of a bibliographical knowledge base. In this setting, I have also worked with Manuel Atencia, Jerome David (both MCF, Grenoble), Michel Chein (PR, Université de Montpellier 2), Michel Leclere, Madalina Croitroru (both MCF , Université de Montpellier 2) to compare different key semantics. This comparison has been published in the International conference ICCS in 2014 (Atencia et al. (2014)).

3.3.2 Motivation

Most of the data linking approaches are based on rules that specify conditions that two data items must fulfill in order to be linked. Linkage rules can be manually defined (Low et al. (2001), Volz et al. (2009), Arasu et al. (2009)), learnt on datasets (Isele and Bizer (2012), Ngomo and Lyko (2012), Nikolov et al. (2012b)) or built from knowledge declared in an ontology (Saïs et al. (2009), Hu et al. (2011), Al-Bakri et al. (2015)). In particular, a key expresses a set of properties whose values uniquely identify every resource of a dataset and thus represent a highly discriminative set of properties that can be used in a linking process. Keys can be used as logical rules to link data when a high precision is needed, or to construct more complex similarity functions. They can also be used by blocking methods to reduce the search space.

Nevertheless, when the ontology represents many concepts and properties, the keys cannot easily be specified by a human expert. Indeed, non composite keys (e.g. ISBN for books or SSN for people) are rare in real data and composite keys are not obvious to specify. For example, is the combination of the properties *birthdate*, *deathdate* and *lastname* sufficient to uniquely identify every person? Furthermore, since data are heterogeneous and generally incomplete, a data linking tool needs keys that involve different sets of properties. Using a rich set of keys, the set of identity links that can be discovered can be more complete. Therefore, we need methods that discover them automatically from the data.

The problem of automatic key discovery has been largely studied in the relational database setting. However, the problem is more complex in the context of the Semantic Web. First, relational databases do not consider semantics (classes, subsumption relation, key inheritance). Second, unlike Semantic Web, multivalued properties are not taken into account in a relational setting. Last, in the Semantic Web context, RDF data may be incomplete and asserting the Closed World Assumption (CWA), i.e., what is not currently known to be true is false, is not meaningful. Hence, heuristics are needed to discover keys in incomplete data.

Key discovery approaches have been proposed recently in the setting of the Semantic Web (Atencia et al. (2012), Soru et al. (2015)). However, these approaches discover keys that do not follow the OWL2 semantics of a key. There are defined to be applicable when a local completeness of data is known (i.e., all the property instances are known for each class instance).

In this work, our aim was to define approaches that can exploit RDF data sources to discover sets of composite keys that follow the OWL2 semantics of a key.

3.3.3 Contributions

We have proposed two automatic approaches that aim to discover keys in RDF data sources. Both approaches discover a set of keys for each instantiated class of each ontology of each considered data source and merge the obtained keys in order to find keys that are valid in all the considered data sources.

- The first approach named KD2R (Key Discovery for Reference Reconciliation) discover keys from datasets for where the UNA is fulfilled. To avoid scanning all the data, KD2R discovers first maximal non keys before deriving the keys. In addition to this, KD2R exploits key inheritance between classes in order to prune the non key search space.

- The second approach named SAKey (Scalable Almost Key discovery) is able to discover set of properties that are highly discriminative. Indeed, this second approach allows the discovery of keys with exceptions in order to learn keys in datasets that may contain erroneous data or duplicates. In this second approach, various pruning strategies have been developed in the non key search and a new algorithm has been proposed to derive keys from non keys. An extension of SAKey called C-SAKey have been proposed to learn conditional keys: keys that are valid in a subpart of the data.

We have also theoretically and experimentally compared keys when different semantics of the keys are considered.

3.3.4 KD2R: a key discovery approach for data linking

Key discovery problem considered in KD2R : Let s_1 and s_2 be two RDF data sources that conform to two OWL ontologies O_1, O_2 respectively.

We consider in each data source s_i the set of instantiated property expressions $\mathcal{P}e_i = \{pe_{i1}, pe_{i2}, \ldots, pe_{iN}\}$. Let $C_i = \{c_{i1}, c_{i2}, \ldots, c_{iL}\}$ be the set of classes in the ontology O_i. Let $\mathcal{M}$ be the set of equivalence mappings between the elements (property expressions or classes) of the ontologies O_1 and O_2. Let $\mathcal{P}e_{1c}$ (resp. $\mathcal{P}e_{2c}$) be the set of properties of $\mathcal{P}e_1$ (resp. of $\mathcal{P}e_2$) such that there exists an equivalence mapping with a property of $\mathcal{P}e_2$ (resp. of $\mathcal{P}e_1$).

The problem of key discovery that we address in this work is defined as follows:

1. for each data source s_i and each class $c_{ij} \in C_i$ of the ontology O_i, such that it exists a mapping between a class c_{ij} and a class c_{ks} of the other ontology O_k, discover the parts of $\mathcal{P}e_i$ that are keys in the data source s_i,
2. find all the parts of $\mathcal{P}e_{ic}$ that are keys for equivalent classes in the two data sources s_1 and s_2 with respect to the property mappings in $\mathcal{M}$.

Optimistic and pessimistic heuristics: The Closed Word Assumption (CWA) can rarely be ensured in an RDF data source. However, to discover keys in a dataset, we theoretically need all the *owl:sameAs* and *owl:differentFrom* links existing in the dataset. Since these links are generally not described in RDF datasets, KD2R discovers keys in datasets for which the Unique Name Assumption (UNA) is fulfilled, i.e., there exists an implicit *owl:differentFrom* link for every pair of instances in the data.

Moreover, discovering keys when some property instances might be missing needs some heuristics to be defined. KD2R uses either a pessimistic or an optimistic heuristic. In these heuristics, for one class instance, we have distinguished instantiated properties from non instantiated properties. In both heuristics, instantiated properties are considered as completely described (e.g if some of the authors of a paper are declared, we consider that they are all declared). Using a pessimistic heuristic, the property for which no value is given may take all the values that appear in the data source. Using an optimistic heuristic, we consider that the not given property values are empty or different from all the values that appear in the data source for this property. These two heuristics lead us to define keys, non keys and undetermined keys. The undetermined keys are considered to be keys in the optimistic heuristic and non keys in the pessimistic heuristic. More precisely, we consider that a set of property expressions is a *key* (*c.f.* definition 1) for a class if for all pairs of distinct instances of this class, there exists an instantiated property expression in this set such that all the values are distinct (objects or literal values).

Definition 1 – Keys. A set of property expressions $k_{s_i.c} = \{pe_1, \ldots, pe_n\}$ is a key for the class c in a dataset s_i if:

$$\forall X\ \forall Y\ ((X \neq Y) \wedge c(X) \wedge c(Y)) \Rightarrow$$

$$\exists pe_j\ (\exists U\ \exists V\ pe_j(X,U) \wedge pe_j(Y,V)) \wedge (\forall Z\ \neg(pe_j(X,Z) \wedge pe_j(Y,Z)))$$

We denote $K_{s_i.c}$ the set of keys of the class c w.r.t the data source s_i.

Example. The following RDF source $s1$ contains the RDF descriptions of four *db*:*Restaurant* instances.

Source $s1$**:**
$db{:}Restaurant(r1), db{:}name(r1,''\ Arzak''), db{:}city(r1,c1),$
$db{:}address(r1,''\ 800\ \ Decatur\ Street''), db{:}country(r1,''\ Spain''),$
$db{:}Restaurant(r2), db{:}name(r2,''\ Park\ \ Grill''), db{:}city(r2,c2),$
$db{:}address(r2,''\ 11\ North\ Michigan\ Avenue''), db{:}country(r2,''USA''),$
$db{:}Restaurant(r3), db{:}name(r3,''\ Geno's\ Steaks''), db{:}country(r3,''\ USA''),$
$db{:}telephone(r3,''\ 884-4083''), db{:}telephone(r3,''\ 884-4084''),$
$db{:}address(r3,''\ 35\ \ cedar\ Avenue''),$
$db{:}Restaurant(r4), db{:}name(r4,''\ joy\ Hing''), db{:}city(r4,c4),$
$db{:}address(r4,''\ 265\ Hennessy\ Road''), db{:}country(r4,''\ China'')$

In this example, $\{db:address\} \in K_{s1.db:Restaurant}$ since the addresses of all the restaurants that appear in the data source $s1$ are distinct.

We consider that a set of property expressions is a *non key* (*c.f.* definition 2) for a class if there exist two distinct instances of this class that share the same values for all the property expressions of this set.

Definition 2 – Non keys. A set of property expressions $nk_{s_i.c} = \{pe_1, \ldots, pe_n\}$ is a non key for the class c in one data source s_i if:

$$\exists X\ \exists Y\ \exists Z_1, \ldots, \exists Z_n\ (pe_1(X, Z_1) \wedge pe_1(Y, Z_1) \wedge \ldots \wedge$$

$$pe_n(X, Z_n) \wedge pe_n(Y, Z_n) \wedge (X \neq Y) \wedge c(X) \wedge c(Y))$$

We denote $NK_{s_i.c}$ the set of non keys of the class c w.r.t the data source s_i.
Example. $\{db : country\} \in NK_{s1.Restaurant}$ since there are two restaurants that are located in the same country (USA) in the data source $s1$.

So, some combinations of property expressions are neither keys nor non keys: a set of property expressions is called an *undetermined key* (*c.f.* definition 3) for a class if it is not a non key and there exist two instances of the class such that the instances share the same values for a subset of the property expressions, and the remaining property expressions are non instantiated for at least one of the two instances.

Definition 3 – Undetermined Keys. A set of property expressions $uk_{s_i.c} = \{pe_1, \ldots, pe_n\}$ is an undetermined key for the class c in s_i if:

- (i) $uk_{s_i.c} \notin NK_{s_i.c}$ and
- $(ii) \exists X\ \exists Y\ (c(X) \wedge c(Y) \wedge (X \neq Y)$

  $\wedge \forall pe_j\ ((\exists Z\ (pe_j(X, Z) \wedge pe_j(Y, Z)) \vee \nexists W\ (pe_j(X, W) \vee \nexists W\ pe_j(Y, W))))$

We denote $UK_{s_i.c}$ the set of undetermined keys of the class c w.r.t the data source s_i.
Example. $\{db{:}country, db{:}city\} \in UK_{s1.Restaurant}$ since it is not a non key and there are two restaurants in the same country(USA) but one of them does not contain any information about the city where it is located.

KD2R overview

The key discovery problem is exponential in the number of properties. A naive solution that would check all the possible combinations of property expressions does not scale: for a class described by 15 properties, the number of candidate keys is $2^{15} - 1$. To reduce the number of computations, we have proposed a method inspired by Sismanis et al. (2006) which first retrieves the set of maximal non keys and then computes the set of minimal keys, based on this set of non keys. Indeed, to make sure that a set of property expressions is a key, we have to scan the whole set of instances of a given class. On the contrary, finding two instances that share the same values for the considered set of property expressions suffice to be sure that this set is a non key.

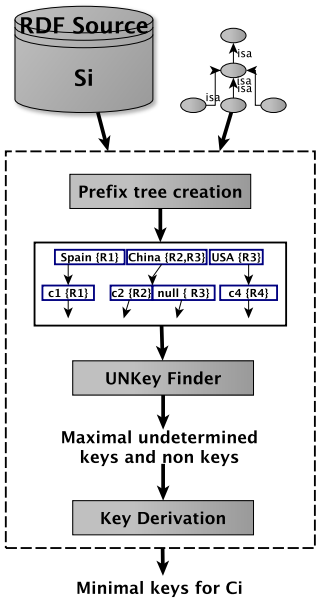
Figure 3.7 presents the main steps of KD2R approach. Our method discovers the key constraints for each RDF data source independently. In

each data source, KD2R is applied on the classes in topologically sorted order. In this way, the keys that are discovered in the superclasses of a given ontology can be exploited when processing their subclasses. For a given data source s_i and a given class c we have defined the algorithm *Key Finder* which aims at finding keys for the class c that are valid in the data source s_i. *Key Finder* starts by building a prefix tree for this class to represent its instances in a compact way (see Figure 3.7(a)). Using this representation, the sets of maximal undetermined keys and maximal non keys are computed (algorithm *UNKey Finder*).

These sets of undetermined keys and non keys are then used to derive the set of minimal keys (*Key Derivation* algorithm). For this step, we have exploited the algorithm proposed by Sismanis et al. (2006) in the relational database context. The main idea of the derivation process is that a key is a set of properties that is not included or equal to any maximal non key or maximal undetermined key.

The obtained keys are then merged in order to compute the set of key constraints that are valid for both data sources (see Figure 3.7(b)) (*Key Merge*). This is done by the selection of the minimal keys that belong to the Cartesian product of the discovered keys.

(a) *Key Finder* for one data source



(b) *Key Merge* for two data sources

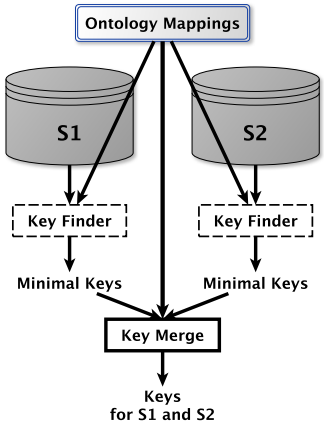


Figure 3.7 – *KD2R: Key Discovery for two data sources*

To illustrate the results that can be obtained by each algorithm, I present the prefix tree, and the results obtained by *Key Finder* for the class *db*:*Restaurant* instances of the data source *s1* described in example 3.3.4. In this example, I suppose that a pessimistic heuristic has been chosen. In the prefix tree 3.8, each level corresponds to a property expression, each node contains a set of cells and each cell contains (1) a property value, (2) a first URI list (objects that share the cell value or such that this value is possible), and (3) a second URI list that stores the objects for which the property has not been instantiated. This second list is used by *UNK Finder* to distinguish

non keys and undetermined keys.

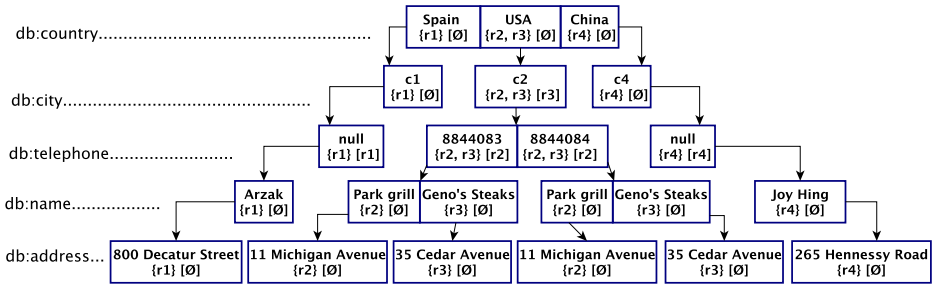


Figure 3.8 – *Prefix-tree for the db:Restaurant class instances (Pessimistic heuristic)*

Using a depth first traversal of this tree and merge node operations that allow to ignore exploring all property combinations, *UNKey Finder* searches for the biggest combination of property expressions having values that are shared by several instances. *UNKey Finder* applies three kinds of pruning strategies during the prefix tree exploration:

- **Key pruning:** when a key is already discovered for a class, then this key is also valid for all its subclasses.
- **Antimonotonic pruning:** if a set of properties is a non key, all its subsets are, by definition, non keys.
- **Monotonic pruning:** if a set of properties is a key then all the supersets of this key are also keys.

In this example, we obtain only one maximal non key and one maximal undetermined key:
$UK_{s1.db:Restaurant} = \{\{db{:}telephone, db{:}city, db{:}country\}\}$
$NK_{s1.db:Restaurant} = \{\{db{:}country\}\}$
The two minimal keys that can be derived from these two sets are $K_{s1.db:Restaurant} = \{\{db{:}address\}, \{db{:}name\}\}$.
Finally, if *Key Finder* is also applied to the restaurants of another data source *s2*, the keys can be merged to obtain keys that are valid in both data sources. For example, if $K_{s2.db:Restaurant} = \{\{db : telephone, db : city\}, \{db : name\}\}$ then the keys obtains for these two data sources are:
$K_{D:Restaurant} = \{\{db : telephone, db : address, db : city\}, \{db : name\}\}$.

Evaluation overview

The approach has been implemented and evaluated on 13 different datasets: 6 datasets that have been used in the Instance matching track of OAEI, 4 datasets that have been collected on the LOD and finally 3 datasets that were provided by partners of the Qualinca project. For these last datasets, the discovered keys have been evaluated by experts but there relevance was not so easy to determine (many composite keys were considered as not always valid). Thus, to evaluate the quality of the discovered keys, we have used them in a linking process on benchmark datasets of the international contest OAEI 2010, 2011 and 2012. The results obtained by N2R using KD2R

keys showed that the use of these keys has led to generate more relevant identity links than when N2R is used without keys (weighted average of the similarities obtained with all properties). Moreover, it has been shown that the results with KD2R keys are similar to the results obtained with keys that have been manually defined by experts. KD2R has been applied on some large DBpedia classes (for instance DBpedia person, 3 332 207 RDF triples). However the experimentations have also shown that when large datasets containing many properties are considered, KD2R cannot scale. In addition, one class of the datasets provided by the Qualinca project was containing duplicates. KD2R was not able to discover keys for this class.

3.3.5 SAKey: a Scalable Almost-Key discovery approach

RDF data, in particular data published on the Web, may contain erroneous information or duplicates. When these data are exploited to discover keys, relevant keys can be lost. For example, if a dataset contains an erroneous social security number (SSN) and if this SSN is associated to another person, this property will not be considered as a key in KD2R. Allowing some exceptions can prevent the system from losing keys. Furthermore, the number of keys discovered in a dataset can be few. However, even if a set of properties is not a key, it can be used to generate many identity links. For example, the telephone number of a restaurant can generally be used to identify a restaurant. Nevertheless, there can be two different restaurants located in the same place sharing phone numbers.

We have proposed a second approach called SAKey that exploits RDF datasets to discover *almost keys*. An almost key is a set of properties that is not a key due to few exceptions. The set of almost keys is derived from the set of non keys found in the data. SAKey can scale on large datasets by applying a number of filtering and pruning techniques that reduce the requirements of time and space.

Key discovery problem considered in SAKey In Fig. 3.9, five films are described by their name, their release date, the language in which they are filmed, their actors and directors.

Dataset D1:
d1:Film(f1), $d1{:}hasActor(f1,''B.Pitt'')$, $d1{:}hasActor(f1,''J.Roberts'')$,
$d1{:}director(f1,''S.Soderbergh'')$, $d1{:}releaseDate(f1,''3/4/01'')$, $d1{:}name(f1,''Ocean's\ 11'')$,
d1:Film(f2), $d1{:}hasActor(f2,''G.Clooney'')$, $d1{:}hasActor(f2,''B.Pitt'')$,
$d1{:}hasActor(f2,''J.Roberts'')$, $d1{:}director(f2,''S.Soderbergh'')$, $d1{:}director(f2,''P.Greengrass'')$,
$d1{:}director(f2,''R.Howard'')$, $d1{:}releaseDate(f2,''2/5/04'')$, $d1{:}name(f2,''Ocean's\ 12'')$
d1:Film(f3), $d1{:}hasActor(f3,''G.Clooney'')$, $d1{:}hasActor(f3,''B.Pitt'')$
$d1{:}director(f3,''S.Soderbergh'')$, $d1{:}director(f3,''P.Greengrass'')$, $d1{:}director(f3,''R.Howard'')$,
$d1{:}releaseDate(f3,''30/6/07'')$, $d1{:}name(f3,''Ocean's\ 13'')$,
d1:Film(f4), $d1{:}hasActor(f4,''G.Clooney'')$, $d1{:}hasActor(f4,''N.Krause'')$,
$d1{:}director(f4,''A.Payne'')$, $d1{:}releaseDate(f4,''15/9/11'')$, $d1{:}name(f4,''The\ descendants'')$,
$d1{:}language(f4,''english'')$
d1:Film(f5),$d1{:}hasActor(f5,''F.Potente'')$, $d1{:}director(f5,''P.Greengrass'')$,
$d1{:}releaseDate(f5,''2002'')$, $d1{:}name(f5,''The\ bourne\ Identity'')$, $d1{:}language(f5,''english'')$
d1:Film(f6),$d1{:}director(f6,''R.Howard'')$, $d1{:}releaseDate(f6,''2/5/04'')$,
$d1{:}name(f6,''Ocean's\ twelve'')$

Figure 3.9 – *Example of RDF data*

In this example, the property $d1$:*hasActor* is not a key for the class *Film* since there exist actors that play in several films. For example, "*G. Clooney*" plays in films $f2$, $f3$ and $f4$. Various notions of key exceptions could have been considered (number of false identity links that could be produced with this key for example). We have considered each film that shares actors with other films as one exception. So, there exist 4 exceptions for the property $d1$:*hasActor*.

Formally, the set of exceptions E_P corresponds to the set of instances that shares values with at least one instance, for a given set of properties P.

Definition 2. (Exception set). Let c be a class ($c \in \mathcal{C}$) and P be a set of properties ($P \subseteq \mathcal{P}$). The exception set E_P is defined as:

$$E_P = \{X \mid \exists Y (X \neq Y) \wedge c(X) \wedge c(Y) \wedge (\bigwedge_{p \in P} \exists U p(X,U) \wedge p(Y,U))\}$$

For example, in D1 of Figure 3.9 we have: $E_{\{d1:hasActor\}} = \{f1, f2, f3, f4\}$, $E_{\{d1:hasActor,\, d1:director\}} = \{f1, f2, f3\}$.
Using the exception set E_P, we have proposed the following definition of a n-almost key.

Definition 3. (n-almost key). Let c be a class ($c \in \mathcal{C}$), P be a set of properties ($P \subseteq \mathcal{P}$) and n an integer. P is a *n-almost key* for c if $|E_P| \leq n$.

This means that a set of properties is considered as a n-almost key, if there exist from 0 to n exceptions in the dataset. For example, in D1, we consider the property $d1$: *hasActor* as a 4-almost key since it contains at most 4 exceptions. By definition, if a set of properties P is a n-almost key, every superset of P is also a n-almost key.

The SAKey approach aims to discover minimal n-almost keys, i.e., n-almost keys that do not contain subsets of properties that are n-almost keys for a fixed n.
As we have already shown in KD2R, an efficient way to obtain keys, is to discover first all the non keys and use them to derive the keys. In SAKey, we have also applied this idea. SAKey derives the set of n-almost keys from the sets of properties that are not n-almost keys. Indeed, to show that a set of properties is not a n-almost key, i.e., a set of properties with at most n exceptions, it is sufficient to find at least (n + 1) instances that share values for this set. We call the sets that are not n-almost keys, $(n+1)$-non keys.

The approach is based on a data structure, called *Imap*, that stores for each property, cluster of instances that share the same property value.

Using this data structure, three main steps are applied:

1. **Preprocessing step.** In this step data are filtered: clusters of size one, and clusters that are included in another cluster for the same property, irrelevant property combinations (single keys and some of combination of properties that cannot be a n-non keys).

2. **n-non key discovery.** In this step, the discovery of maximal (n+1)-non keys (Algorithm *nNonKeyFinder*) is done.

Table 3.3 – *Initial map of D1*

$d1$:*hasActor*	{{$f1$, $f2$, $f3$}, {$f2$, $f3$, $f4$}, {$f1$, $f2$}, {$f4$}, {$f5$}, {$f6$}}
$d1$:*director*	{{$f1$,$f2$,$f3$}, {$f2$, $f3$, $f5$}, {$f2$, $f3$, $f6$}, {$f4$}}
$d1$:*releaseDate*	{{$f1$}, {$f2$, $f6$}, {$f3$}, {$f4$}, {$f5$}}
$d1$:*language*	{{$f4$, $f5$}}
$d1$:*name*	{{$f1$}, {$f2$}, {$f3$}, {$f4$}, {$f5$}, {$f6$}}

3. ***n*-almost key derivation.** In this step, a derivation of n-almost keys from the set of(n+1)-non keys (new Algorithm for *keyDerivation*) is taking place.

SAKey: Evaluation overview

In contrast to KD2R, our system is able to scale when data are large. Our extensive experiments show that SAKey can run on millions of triples. Even when many exceptions are allowed, SAKey can still discover keys efficiently. Moreover, the experiments demonstrate the relevance of the discovered almost keys: when few exceptions are allowed, the recall increases significantly and the precision is not so affected by this choice.

C-SAKey: a conditional Scalable Almost Key discovery approach

For some datasets, ontology classes are very general and only few keys can be discovered. In C-SAKey, we have proposed to discover conditional keys that are valid only for a subset of the instances of a class.

OWL2 allows to declare a key for a class expression $c \wedge cd$ where c represents a class and cd a condition. To express conditions on property values, the constructs *owl:DataHasValue*, noted $dhv(p, value)$, or *owl:ObjectHasValue* can be used [4].
The semantics of a conditional key (($c \wedge cd$) ($p_1, \ldots, p_n$)) is the following:

$$\forall X, \forall Y, \forall Z_1, \ldots, Z_n, \wedge c(X) \wedge c(Y) \wedge cd(X) \wedge cd(Y) \bigwedge_{i=1}^{n} (p_i(X, Z_i) \wedge p_i(Y, Z_i)) \Rightarrow X = Y$$

We have chosen to extract the non keys using SAKey and to use this set to reduce the search space for the conditional keys. Indeed, the properties that are involved in a minimal conditional key are included in a non key. Then, to construct the conditions two choices can be made:

- In the first case, an expert chooses among the properties of the non key, the properties that will appear in the condition. For example, the expert may be interested in discovering the keys for people depending on research labs where they work. Obviously, the name of person is not a key in every research lab.
- In the second case, property values are selected depending on the size of the class that can be constructed using this property value. For example, let us consider a dataset describing cities where these cities

[4] see http://www.w3.org/TR/owl2-syntax/#Class_Expressions for more details.

clustered using the region they are located in. Since the size of the cluster region can be significantly large it may be interesting to discover keys for cities for every specific region.

A first experimentation has been conducted on the INA dataset, dataset for which no keys can be found when exceptions are not allowed. The results have shown that C-SAKey can discover keys for this class and that these keys can vary depending on the constructed class expression.

3.3.6 Different Key semantics for RDF datasets: a theoretical and experimental comparison

We have seen that different approaches have been proposed to automatically induce keys from RDF datasets, and then exploit discovered keys for datasets cleaning and interlinking (Pernelle et al. (2013b), Atencia et al. (2012), Soru et al. (2015)). Nevertheless, these keys may have different semantics. In (Pernelle et al. (2013b), Symeonidou et al. (2014)), each pair of instances that share at least one value for each property of the key should be considered as referring to the same entity. In (Atencia et al. (2012), Soru et al. (2015)), the authors consider that each pair of instances should coincide for all the property values to be considered as referring to the same entity. This last semantics can be interesting when one can guarantee that local completeness assumption is fulfilled. We have formalized these different notions of a key in the context of RDF dataset cleaning and interlinking, and we have given some experimental results of these different key notions for both problems of cleaning and interlinking.

More precisely, we have considered two notions of keys: *S*-key and *F*-key. *S*-key roughly corresponds to the `hasKey` axiom of OWL2 and the notion of a key used by Pernelle et al. (2013b), Symeonidou et al. (2014). In Atencia et al. (2012), Soru et al. (2015), a trade-off between *S*-key and *F*-key is considered.

Definition 4 (*S*-key). The *S*-key $\{p_1, \ldots, p_n\}$ for a class expression C is the rule defined as follows:

$$\forall x \forall y \forall z_1 ... z_n (C[x] \wedge C[y] \wedge \bigwedge_{i=1}^{n} (p_i(x, z_i) \wedge p_i(y, z_i)) \rightarrow \ x = y)$$

where $C[.]$ is a class expression.

The definition of the `hasKey` axiom given in OWL 2 enforces the considered instances to be named (*i.e.,* they have to be URIs or literals, but not blank nodes).

S-key and OWL 2 `hasKey` do not require that two class instances coincide on all values of the key properties to be equal: it suffices to have at least one pair of values that coincide for all p_i to decide that x and y refer to the same entity. However, in case of not functional properties that represent a full list of items (e.g., a list of authors of a given paper, a list of actors of a given movie) it can be more meaningful to consider the fact that the instances should coincide for all the property values. We call this second semantics *Forall-key* and we give its formal definition in the following.

Definition 5 (*F*-key)**.** The *F*-key for a class C is the rule defined as follows:

$$\forall x \forall y (C[x] \wedge C[y] \wedge \bigwedge_{i=1}^{n} (\forall z_i (p_i(y, z_i) \rightarrow p_i(x, z_i)) \wedge$$

$$(\forall w_i(p_i(x, w_i) \rightarrow p_i(y, w_i))) \rightarrow (x = y))$$

Illustrative Example. Figure 3.10 and Figure 3.11 help to compare the two notions of a key described above in a scenario of datasets cleaning. First, consider the RDF graph $G1$ shown in Figure 3.10. If the datatype property *myLab*:*hasEMail* is declared to be a S-key of the class *myLab*:*Researcher* then the two researchers *myLab*:*ThomasDupond* and *myLab*:*TomDupond* must be the same, since they share "thomas.dupond@mylab.org"[5]. If the datatype property *myLab*:*hasEMail* is declared to be a F-key of the class *myLab*:*Researcher*, then we cannot infer that these two researchers are the same. In this case, declaring *myLab*:*hasEMail* as a S-key is more appropriate. Now, consider the graph $G2$ depicted in Figure 3.11. It seems

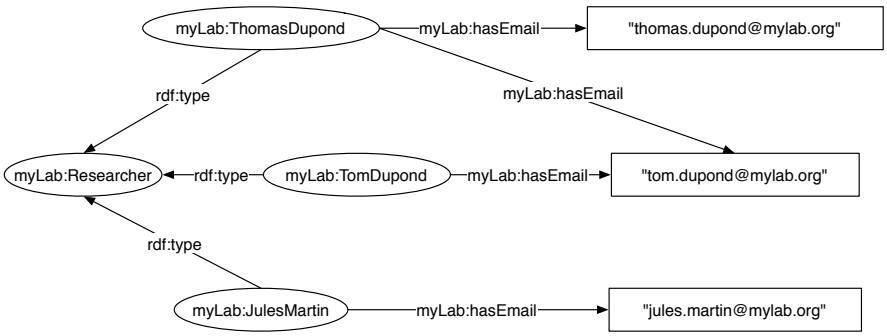


Figure 3.10 – *A RDF graph* $G1$

not to be appropriate to declare that the object property *myLab*:*isAuthor* is a S-key for *myLab*:*Researcher*. Indeed, this would lead us to infer that *myLab*:*TomDupond* and *myLab*:*JulesMartin* are the same just because they have been coauthors in the paper *http://papersdb.org/conf/145*. On the other hand, it is unlikely that different researchers are authors of exactly the same publications. If we declare the object property *myLab*:*isAuthor* as a F-key for *myLab*:*Researcher* then we can infer only that the two researchers *myLab*:*ThomasDupond* and *myLab*:*TomDupond* are the same person. Indeed, using this F-key would not lead us to equate *myLab*:*TomDupond* and *myLab*:*JulesMartin* because the latter is not an author of *http://papersdb.org/conf/26*.

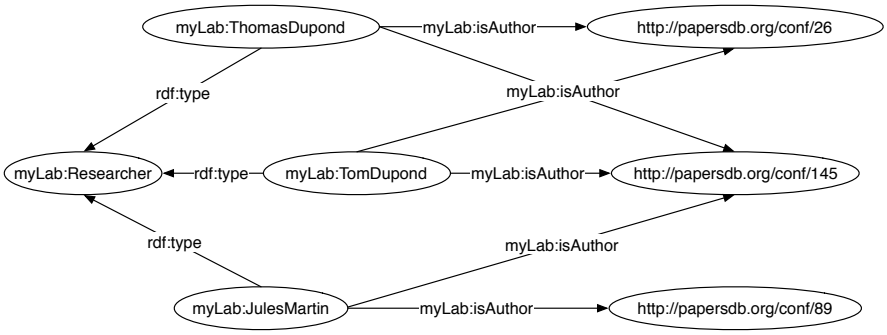


Figure 3.11 – *A RDF graph* $G2$

We have described the relationships between the interpretations satisfying S-keys and F-keys. These relationships depend on the cardinality of the

[5]This is solved by adding myLab:ThomasDupond owl:sameAs myLab:TomDupond to the graph.

properties involved in the keys (no values, one value, multiple values). The most important property of the following proposition is that the *S*-key notion is more restrictive than the *F*-key notion when one considers interpretations in which key properties are always valued.

Proposition 1. *Let S-K and F-K be respectively the* S*-key and the* F*-key associated with* $(C, (p_1, \ldots, p_n))$.

1. *For any interpretation I such as for any* $i = 1, \ldots, n$ *there is at most one element* $c \in C^I$ *with* $p_i^I(c) = \emptyset$, *one has: if I is a model of S-K then I is a model of F-K (i.e., if* $I \models$ *S-K then* $I \models$ *F-K).*

2. *For any interpretation I such as for any element* $c \in C^I$ *and any property* p_i *one has* $card(p_i^I(c)) \leq 1$, *then if I is a model of F-K then I is a model of S-K (i.e., if* $I \models$ *F-K then* $I \models$ *S-K).*

3. *For any interpretation I such as for any element* $c \in C^I$ *and any property* p_i *one has* $card(p_i^I(c)) = 1$, *then I is a model of S-K iff I is a model of F-K (i.e.,* $I \models$ *F-K iff* $I \models$ *S-K).*

Theoretically, a key K that is discovered on a dataset ensures the logical satisfiability of the considered RDF dataset enriched by K. However, discriminability measures can be defined.

Experimental comparison. We have provided experimental evaluations of the different semantics of keys for both interlinking and cleaning scenarii. Results show that learning *F*-keys from data is not suitable when properties are not almost fully instantiated. The *SF*-key variant allows to fix this problem by relaxing the equality constraint when instances have no value for a property. When applied to interlinking or cleaning tasks, *S*-keys and *SF*-keys can have almost the same relevance in term of recall and precision. *SF*-keys and *F*-keys seem to be more robust than *S*-keys when instances are suppressed. Each semantic of key has its own advantages and we think that it could be interesting to define and discover hybrid keys (i.e., keys composed of *F* properties and *S* properties) when data knowledge and/or ontology axioms can be used to decide how properties can be handled.

3.3.7 Summary

We have designed approaches capable of discovering OWL2 keys in RDF data. These RDF data can be numerous, incomplete and they can contain errors or some duplicates. Theoretically, a key discovery approach cannot obtain meaningful results when the complete set of sameAs statements is not declared in the explored dataset. Therefore, we have exploited datasets for which the UNA is fulfilled. Moreover, in the setting of OWA, since not all the property instances are known, we have proposed two heuristics to interpret the potential absence of information. To discover keys in different datasets that conform to distinct ontologies, we have proposed a strategy in which keys are learned separately in each dataset. Then, mappings between classes and properties are considered in order to merge the discovered keys. The proposed merging operation computes keys that are valid in every dataset. The first designed approach, KD2R, was not able to deal with data containing

duplicates and errors. Therefore, we have proposed SAKey, an approach that can discover sets of properties that are not keys due to few exceptions. A preliminary work has also been proposed to compute conditional keys: keys that are valid for a given subset of class instances.

Since other emerging approaches were based on a different semantics of a key, we have compared these two semantics theoritically and experimentally. Our conclusion is that hybrid keys could be very relevant when some of the data properties are known to be local complete.

3.4 A logical approach to detect invalid Identity links

3.4.1 Context

I have been working on an invalidation approach with Laura Papaleo (post-Doc researcher), Fatiha Saïs and Cyril Dumont (engineer) in 2014. The approach has been described in the proceedings of the International Conference EKAW (Papaleo et al. (2014)). This work has been developed in the setting of the Qualinca project.

3.4.2 Motivation

Most of the RDF links connecting resources coming from different data sources are *RDF identity links*, and are defined using the *owl:sameAs* property. Unfortunately, as argued recently within the research community (de Melo (2013)), many existing identity links do not reflect such genuine identity.

Problems arise both in cases in which sameAs is automatically discovered by a data linking tool erroneously, or when users declare it but meaning something less 'strict' than the semantics defined by OWL. Can I declare a sameAs between the resource that refers to a person twenty years ago and a resource that refers to this person today? Can I declare a sameAs between two book editions? Thus, it is becoming important to develop means of linking quality assurance. The study of the quality of identity links may be particularly useful in applications that want to consume Linked Data as well as in Semantic Web frameworks dedicated to data linking or data integration. In such applications, transferring properties across URIs that are linked by a sameAs statement can only be done if a strict form of identity is guaranteed.

The task of discovering erroneous identity links is rather novel. However, a small number of attempts exist. In (Halpin et al. (2011)) the authors studied the problem of the quality of RDF identity links from a general point of view, making observations about the varying use of *owl:sameAs* in Linked Data. They proposed an ontology called the Similarity Ontology (SO) that aims at better classifying the different levels of similarity between items in different data sources. However, the quality evaluation of the *owl:sameAs* links is performed manually, in an Amazon Mechanical Turk experiment. In (de Melo (2013)), the author illustrates how to assess the quality of *owl:sameAs* links, using a constraint-based method. In this work, the transitivity of sameAs and the Unique Name Assumption is used to detect inconsistencies and a relaxation algorithm is used to propose erroneous sameAs. However, as claimed by the author himself, it could be important to include advanced similarity measures and the evaluation of more properties.

Let us consider a very simple example: we have two books b_1 and b_2 both described using two data-type properties *isbn* and *pages*. In order to infer $sameAs(b_1, b_2)$, an application or a user has supposed that it is sufficient to check if the values of *isbn* are equal, since isbn is inverse functional. However if the values of the mono-valued property pages are not equivalent, one can detect a conflict. Thus, once a sameAs statement exists in the knowledge base, it could be interesting to analyze different properties (not only keys). To do that, we need to know that the property pages is functional (axiom of the ontology) and that the number of pages are different (lexical information). The problem we have addressed is to check if a sameAs statement sameAs(x,y) can be logically invalidated when ontology axioms and lexical knowledge are taken into account.

3.4.3 Contributions

We have proposed a logical method to detect invalid sameAs statements, by looking at the descriptions associated to the instances. This logical method relies on ontology axioms and lexical resources and we have supposed that, in case of multiple heterogeneous data sources, mappings between properties are provided. Our approach is local, in the sense that, we build a contextual graph 'around' each one of the two resources involved in the sameAs statement and we exploit the descriptions provided in these contextual graphs. The construction of a contextual graph is based on properties that have specific characteristics: functionality and local completeness.

3.4.4 Detection of invalid Identity links

Problem statement

Our approach relies on building two contextual graphs, for two resources x and y respectively and on reasoning on the assertions contained in these two graphs. More precisely, the building blocks of the problem are the following:

- An RDF graph G
- two resources x and y, such that x, y belong to G
- the triple $\langle x, owl : sameAs, y\rangle$ (or $sameAs(x, y)$) belonging to G
- a selected set of properties P in G
- a value n representing the depth of the contextual graphs
- the contextual graphs $G_{\{n,x,P\}}$ and $G'_{\{n,y,P\}}$ for x and y

The problem becomes the evaluation of the following rule:

$$G_{\{n,x,P\}} \wedge G'_{\{n,y,P\}} \wedge sameAs(x, y) \Rightarrow \bot$$

Contextual graph

The construction of the contextual graphs depends on the properties we select and the value n. Indeed, in complex RDF graph, which can combine data coming from multiple data sources, limiting the depth of a contextual graph can prevent from using not relevant piece of information which can eventually confuse the validation process.

Given an RDF graph G, a node s in G, given a set P of properties defined for G, a *Property-based walk of length n* $w_{\{n,s,P\}}$ is basically a path in the RDF graph without cycle and of length n, involving $n+1$ node, n resources defined by URIs and 1 node as a literal.

A *m-degree contextual graph* for a resource s can then be defined as:

Definition 6. *m*-degree Contextual Graph $G_{\{m,s,P\}}$
Given and RDF graph G and a node $s \in G, s \in U$, an integer number m and a set P of properties defined for G, a m-degree Contextual Graph $G_{\{m,s,P\}}$ for s is a sub-graph of G such that every node $v_i \in G_{\{m,s,P\}}$ belong to a property-based walk of length n, with $n \leq m$.

A *m-degree contextual graph* for a resource s can seen as a subset of knowledge pertinent to s, bounded by a selected set of predicates P.

In Figure 3.12, I show an example of a contextual graph extracted for $r2$ (circles identify resources with URI and rectangles represent literals). In this example, a value $n = 2$ has been selected. The set of properties P has been defined as $\{phone\_number, has\_address, city\}$.

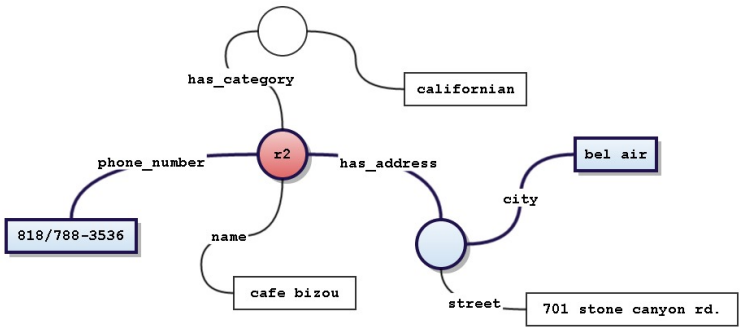


Figure 3.12 – *An instance of restaurant in a OAEI dataset. Given the functional properties phone_number, has_address and city, a contextual graph of degree 2 is depicted*

Properties Selection

We chose to use functional properties and those properties declared as local complete.

In our approach, taking into consideration functional properties, we basically add the following rules for every property p_i, p_j, p_k in the contextual graphs we are considering.

$$R1_{FDP} : sameAs(x,y) \wedge p_i(x,w_1) \wedge p_i(y,w_2) \rightarrow synVals(w_1,w_2)$$

$$R2_{FOP} : sameAs(x,y) \wedge p_j(x,w_1) \wedge p_j(y,w_2) \rightarrow sameAs(w_1,w_2)$$

$$R3_{IFP} : sameAs(x,y) \wedge p_k(w_1,x) \wedge p_k(w_2,y) \rightarrow sameAs(w_1,w_2)$$

Note that $R_{1_{FDP}}$ is for data-type properties and $R_{2_{FOP}}$ and $R_{3_{IFP}}$ are for object-type properties. *synVals* is a predicate that expresses that two literals are synonyms. Given a property p in the graph G, the knowledge of p being a functional property can be already present among the assertions in G or derived after, collecting knowledge from experts or gathering it externally (existing ontologies, additional assertions on the Web and so on.)

The closed-world assumption is in general inappropriate for the Semantic Web due to its size and rate of change (Heflin and Muñoz-avila (2002)). But in some domains and specific contexts, local-completeness for RDF properties could be assured.

For example, the authors of a publication is generally a multi-valued local complete property. When a predicate is local complete, it should be declared *closed* in the specific knowledge base, making a local completeness assumption.

The Invalidation Approach

Given G the initial RDF graph. Given $sameAs(x, y)$ the input sameAs statement to validate. Let F be a set of facts, initially empty, and L the set of literals for G.

1. Build a set F_1 of $\neg synVals(w_1, w_2)$, for each pair of semantically different w_1 and w_2, with $w_1, w_2 \in L$.
2. Choose a value n indicating the depth of the contextual graphs
3. Build the contextual graphs for x and y considering functional properties and local complete properties
   - For all the functional properties $p_{i_{FP}}$ add the relative set of RDF facts to F, considering the rules $R_{1_{FDP}}, R_{2_{FOP}}, R_{3_{IFP}}$ in Section 3.4.4.
   - For each $p_{i_{LC}}$ that falls in the contextual graphs and fulfills the local completeness (i.e., $R_{4_{LC}}$ is declared), add to F a set of facts in the form $\neg p_{i_{LC}}(s, w)$ if w is different to all the w' s.t. $p_{i_{LC}}(s, w')$ belongs to F, using F_1. Note that $w, w' \in L$.
4. Apply iteratively unit resolution until saturation using $F \cup CNF^6\{R_{1_{FDP}}, R_{2_{FOP}}, R_{3_{IFP}}, R_{4_{LC}}\}$.

The set of $\neg synVals(w_1, w_2)$ can be obtained using different strategies: syntactical similiarities or external lexical resources. For example, normalized dates or years that are syntactically distinct are not equivalent.

Evaluation overview

We have performed experiments for assessing the quality of the set of sameAs statements computed by different linking methods, respectively presented in Saïs et al. (2009), Symeonidou et al. (2014) and Yves et al. (2009). All these methods have produced results on the Person-Restaurants data test available for the instance matching contest OAEI 2010.

[6]CNF: Conjunctive Normal Form

To build the $\neg synVals$ (set F_1) for the values of the properties selected, we did a normalization of the values. For example, for *phone_number*, we removed all the additional characters (e.g. '/', '-', and so on), leaving only the numbers.

Our results showed that, when our invalidation tool is applied after one of the linking method, the precision of each tool can be improved. For Symeonidou et al. (2014) we pass from a precision of 95.55% to 98.85%, for Saïs et al. (2009) from a precision of 69.71% to 95.19% and finally for Yves et al. (2009) from a precision of 90.17% to 100%.

However, we were not able to classify as invalid some of the discovered sameAs which, with respect to the gold standard were erroneous. For example, two restaurants share the same phone number and the same city. They even share the same street name. So an inconsistency cannot be detected. Most probably, they represent the same commercial site providing different services. In addition we have classified as 'wrong sameAs' some statements which, with respect to the gold standard, are in fact correct sameAs. In these examples, the two restaurants have a different phone number or a different city (or both). This could mean that some data are erroneous or that some locations are described more or less accurately. In any case, the approach can highlight the problematic sameAs to the user (expert) and ask for confirmation or correction.

Summary

Recent research discussions within the Linked Data community have shown that the use of *owl:sameAs* may be incorrect. We designed a logical evaluation method which relies on the descriptions associated to the resources involved in the sameAs statement. Given a sameAs statement $sameAs(x, y)$ in a RDF graph G, our method analyzes the functional properties and the properties defined as local complete: it builds a contextual graph for each resource and can detect some anomalies due to the existence of distinct property values.

3.5 Conclusion

In this chapter, the presented approaches aim to enrich RDF datasets with identity links or to improve their quality.

We have defined a global and ontology-based approach, named LN2R, that can combine two steps. The first step exploits axioms declared in the ontology or knowledge known on the data sources to infer correct (non) reconciliation decision. The second step exploits ontology axioms to construct an equation system that is used to compute similarity scores and detect probable identity links. This approach has obtained results that were comparable to those obtained by supervised approaches. However, it has some limitations. First, to be applicable, the approach required data to conform to the same ontology. At that period, wrappers were supposed to be defined to translate data to a single target vocabulary defined in a global schema. Now, in the context of the Linked Data, this data integration architecture is less relevant. Second, a global approach is more "informed" since reconciliation decisions can be propagated to other data item pairs. However, this propagation is time-consuming. To scale, I think that such a global approach needs to be combined

with blocking methods and local approaches. Last, to generate relevant identity links, this method relies on rich ontologies in which disjunctions, (inverse) functional properties or composite keys are defined. However, such axioms are rarely declared in existing ontologies.

A Key discovery approach such as KD2R or SAKey can be exploited to enrich ontologies with some of the needed axioms. This approach can be used even if the schema is not available: keys are then discovered for a virtual single top class. Furthermore, when the two considered data sets are described using different properties, KD2R or SAKey can be applied on the subset of properties of the first dataset that have been previously mapped to properties that are used in the second dataset.

To improve the quality of identity links that are generated by automatic tools, we have proposed a first logical approach that is also based on ontology axioms. We have shown that this approach can detect invalid links. However, this approach needs to be extented to quantify and vizualize the conflicts that can be found for an invalid sameAs. Furthermore, when two resources represent the same object but at different levels of abstractions, or described in different contexts, approaches that can re-qualify such "wrong" sameAs statement are needed.

Perspectives

4

This HDR thesis has presented several of my main contributions obtained in the last 12 years to the domain of data integration and more particular in enriching data semantically both on using annotation strategies and data linking approaches.

More generally, challenges in the context of Web data semantics are still numerous. The Web is an open medium in which everybody can publish data on the Web. As the classic document Web, the Web of Data contains data that can be outdated, conflicting, or intentionally wrong. Furthermore, RDF data that are automatically extracted from more or less syntactically structured sources may contain misrepresentated information. Right now, depending on the application requirements, either data integration approaches deal with inaccurate, incomplete or uncertain data, either they try to assess the quality of Web data and determine the subset of the available data that should be treated as trustworthy.

However, for the semantic vision of the web of data to become a reality, more efforts should be invested in improving data quality. Quality issues for masses of data have to be considered even if the difficulty lies in dealing with the enormous speed at which the Web of data grows. Many complementary research directions are explored to improve data quality such as the definition and validation of structural constraints on RDF graphs, interactive systems that allow domain experts to verify and transform data, use of provenance information. One of the possible ways to improve data quality is to enrich them semantically. In the context of Linked Open Data, ontologies are often incomplete or simply not available. So, approaches are needed to automatically generate meta information about existing datasets or to enrich available ontologies with new concepts, properties, axioms or semantic mappings with other ontology elements. High-quality data descriptions are also required for enriching knowledge bases and high-quality cross-datasets links are necessary to really take advantage of the available datasets.

In this context, the main two research directions I am interested in following in the next three to five years are described in the next two sections: *Discovering high-quality keys for evolving and heterogeneous datasets* (4.1), and *Managing the quality of identity links* (4.2).

4.1 Discovering high-quality keys for evolving and heterogeneous datasets

In this line of research, my aim is to consider new problems of research related to key discovery and directly associated to three points.

First, one of the intrinsic features of Web Data (LOD) is the continuing evolution of its content, including changes daily applied to the data and their corresponding vocabularies (ontologies). Cross-domain ontologies, such as Yago and DBPedia, have evolved extensively since they were first published. For instance, it has been shown that when DBPedia v3.8 is compared to DBPedia v3.7 (49 M of triples), more than five million of property instances are added while more than three million are removed (Roussakis et al. (2015)). The authors have also shown that data-level changes are also numerous for domain-specific knowledge bases such as FMA (experimental biological results) or EFO (Experimental Factor Ontology). Major data integration tasks including synchronization of local data warehouses, maintenance of data linking or data fusion results, or visualization are highly impacted by such modifications. Making the tools performing such data integration tasks aware of the changes and able to update their results is of paramount importance in such an evolving context. Key discovery approaches have only been recently proposed but yet the problem of updating keys has not been properly addressed. This is the problem we propose to work on (section 4.1.1).

Second, there is an increasing number of numerical RDF datasets available today, particularly within scientific RDF datasets. In such a context, key discovery cannot be performed anymore using classical approaches: the identification of data described by numerical properties is more difficult to assess, interpreting the differences between values may not be relevant, and existing key quality criteria are not always adapted. This is the second problem we propose to address (section 4.1.2).

Third, Web semantic data is generally incomplete and data of interest is often described in heterogenous data sources that do not always share a large number of properties. So, generating a larger set of keys would help to improve the results that can be obtained by data linking approaches. The solution we propose to investigate is to build a common framework for key discovery able to consider both keys and referring expressions to augment the set of candidate keys and thus augment the number of identity links that can be detected (section 4.1.3).

4.1.1 Updating keys when data evolve

Many approaches have been defined to detect, represent or assess the impact of changes in Web data, in particular when changes affect the schema level (Zablith et al. (2015)). However, existing key discovery approaches are not designed to efficiently update a discovered set of keys when data changes are detected (Pernelle et al. (2013b), Symeonidou et al. (2014), Soru et al. (2015), Atencia et al. (2012)). Even if pruning strategies can be used to optimize either time or space complexity, discovering a complete set of minimal keys in each dataset is #P-hard (Gunopulos et al. (2003)).

Thus, we plan to maintain up to date the set of (almost-)keys that can be discovered from evolving datasets. We will first study how data changes can be detected and represented in this context. Changes at the schema level (ontology classes, properties) or at the data-level (resources, resource typing, property assertions) can affect the set of valid keys. As in (Roussakis et al. (2015)), we will exploit the data to detect and represent simple (ex: a triple is

added) or complex changes (ex: a class extension is now empty) that may lead to modify the set of valid keys. Then, we will develop new algorithms that exploit change representation to update a set of keys without recomputing it from scratch.

4.1.2 Discovering keys with numerical properties

As previously explained, if we treat numerical data as simple strings in a key discovery approach, we will potentially discover a lot of naive keys. We have initiated a new collaboration with INRA Montpellier on a dataset that describes wines obtained from the Pilotype project[1]. Such a dataset contains a set of numerical values regarding different chemical components that give the flavour of wines. In this application setting, the challenge lies in discovering keys to be used to automatically detect flavour complementarity, unknown from the experts, that allows to distinguish various wine sorts.

To deal with these numerical data, one solution would be to apply property-specific preprocessing steps to convert numerical data into symbolic data. In such a context we plan to group numerical values by following statistical methods. Quantiles are cutpoints dividing a set of observations (data items) into equal sized groups. By using quantiles, we can reduce the number of values and potentially decrease the number of naive keys. Different strategies for computing the quantiles need to be compared (Hyndman and Fan (1996)). Choosing the appropriate size of groups can play a very significant role in the obtained results.

Besides, since the size of the datasets that represent experimental results may not be big enough to discover only relevant keys, various quality criteria are needed to filter the obtained results. Different quality criteria inspired by classical criteria defined in data mining have been exploited (like support and discriminability (Atencia et al. (2012), Symeonidou et al. (2014))). However such criteria cannot always been considered for numerical data. The support is not relevant when all the data are described by the same set of properties, while the discriminability (or number of exceptions) is not so relevant when data are not numerous. Nevertheless, interesting data characteristics can be exploited such as the value distribution and the correlations that can be found between numerical properties. We thus plan to design new quality measures to be defined in the context of numerical datasets that represent experimental results.

4.1.3 From potentially conditional keys to referring expressions

Several strategies may be considered to augment the number of minimal key discovered.

On the one hand and until now, we have developed several approaches to discover OWL keys in RDF datasets. KD2R discovers keys that are valid in several datasets. With SAKey, exceptions are allowed to discover discriminative properties even when erroneous data or duplicates are described in

[1]http://www.qualimediterranee.fr/projets-et-produits/consulter/les-projets/theme-1-agriculture-competitive-et-durable/das2-tic-chaine-alimentaire/theme-1-developper-une-agriculture-competitive-et-durable/das-2-contribution-des-tic-a-la-chaine-alimentaire-en-amont/pilotype

the datasets (e.g. the lab and the firstname is a key for a researcher when 20 exceptions are allowed). To characterize subsets of class instances for which a key is valid, we have proposed to discover conditional keys (C-Sakey) (e.g. the lab and the first name is a key for a researcher that works in France). For this last approach a rather naive algorithm has been implemented to see if relevant conditional keys can be discovered.

One the other hand, some instances have highly discriminative values for a set of properties (or property paths) that are not discriminative for other instances. For example, for a person, the position and the last name are not sufficient to identify a person, but when two persons are named *Obama* and are president of a country, we can say that the two descriptions refers to the same person. These particular keys such that the key support is equal to one (i.e. the rule can be instanciated by only one instance), and such that all the key properties can be valued correspond to the notion of referring expressions. The problem of Reference Expression Generation (REG) has been largely studied in Natural Language Generation (Dale and Reiter (1995)). In this context, the aim is to generate phrases that uniquely identify one domain entity (ex : *the woman with a red hat*). Such algorithms of phrase generation are generally based on simple forms of Knowledge Representation (attribute-value pairs) and find a set of attribute-value pairs whose conjunction is true for the considered domain entity but false for any other entities of the knowledge base. Other approaches have studied this notion in more expressive logical-based settings (Ren et al. (2010), Croitoru and van Deemter (2007)) and it has been shown in (Hu et al. (2011)), that such instance-based discriminative properties can be useful in a data linking setting. Thus we aim at generalizing our work to encompass referring expressions, that we want to consider as a new kind of keys.

So, classically, we have on one hand (conditional) key discovery approaches and on the other hand approaches that discover referring expressions. Our originality will lie on designing a general framework in which both types of rules will be generated. In this framework, the results obtained for keys will be exploited to prune the search space for conditional keys. Then conditional keys will be used to prune, to their turn, the search space for referring expressions. Indeed these rules can be ordered and only minimal rules are needed. Intuitively, if the first name and position is not a key for a person, a conditional key that would express that a first name is a key for persons that are president is relevant. In a same manner, if neither the first name for persons that are presidents nor the position of persons that are named Obama is a key, then the referring expression that only contains constants (the person such that its first name is Obama and its position is president) can be relevant. In this global approach, the key support cannot be used as a criterion to filter unrelevant keys since the support of referring expression is equal to one. Discovering the minimal reference expressions is expensive. So, efficient strategies are needed to enrich a set of key rules with referring expressions that are the most relevant as possible for a data linking process.

With INRA Montpellier and Telecom Paris Tech, we are currently studying how to define more efficient strategies to discover minimal conditional keys. We aim to design an algortihm that relies on the non keys that can be

discovered by SAKey. Then we will define a general framework in which these approaches can be exploited to discover minimal referring expressions.

4.2 Managing the quality of Identity links

Identity links that are automatically generated by data linking tools can be inconsistent. (Halpin et al. (2011)) have shown that 37% of 250 randomly chosen sameAs triples are declared as erroneous by a set of five expert judges. This problem motivates the design of data & linking quality assurance strategies. Such strategies would be useful in applications that want to consume Linked Data or in Semantic Web frameworks dedicated to data integration.

First, approaches that can be used to detect that some sameAs links lead to inconsistent knowledge bases can propose to automatically remove constraint violations from the knowledge base (de Melo (2013)). When high-quality results are needed, a crowdsourced evaluation could be conducted to validate links (Halpin et al. (2011)). Another direction is to design approaches that can detect invalid sameAs statements and that provide explanations that can help experts to correct these erroneous statements. This is the first problem we want to work on (see 4.2.1).

Second, the identity links that have been manually or automatically asserted sometimes reflect weak identity links that cannot be asserted using a sameAs statement. In (Halpin and Hayes (2010)), four distinct uses of *owl* : *sameAs* are presented and discussed. However, as it has been claimed by (de Melo (2013)) *"there is no universal aggreed upon way of determining which properties should count as salient in determining near-identity"*. The representation of weak identity links must be guided by ontologies, expert knowledge and application requirements. This is the second problem we want to address (see 4.2.2).

Third, ontologies can be semantically enriched with rules that can help to make data more complete or to detect erroneous data. Furthermore, rules that are automatically discovered can also help domain experts to better understand and analyse data. Approaches that can discover horn rules in RDF data have been designed (Galárraga et al. (2013; 2015)). The authors have shown that the discovered rules can be used to predict relevant missing values but the length of the discovered rules must be bounded. In scientific domains, experts can be interested in complex but pattern-based rules. This is the third problem we want to address (see 4.2.3).

4.2.1 Assessing the quality of identity links

The task of detecting erroneous identity links is rather new. The logical or constraint-based approaches that have been proposed are based on transitivity, UNA, local completness, and/or ontology axioms (disjunctions or functional properties) (de Melo (2013), Papaleo et al. (2014)). The first results have shown that the precision of data linking tool can be improved when such approaches are applied on the sets of identity links they can discover. However, if we want an expert to correct the knowledge base when erroneous links are detected, information that can help him to understand why an identity link is erroneous must be provided.

Recently, we have begun to study with LIRMM (University of Montpellier) how the argumentation theory could be used to better understand why an identity link leads to an inconsistency. As it has been said in chapter 3, some identity links can be logically inferred thanks to ontology axioms (e.g key properties, transitivity, ...) but they can lead the knowledge base to be inconsistent when two data descriptions contain 'incompatible' elements. The aim is to interact with experts thanks to an explanation dialog where argument-based explanations can be provided.

Argumentation theory is a well-known approach that can be used to deal with inconsistent knowledge (Dung (1995)). Using this theory, arguments can be constructed from inconsistent knowledge bases, attacks can be identified between arguments and some of the arguments and conclusions can be preferred depending on a chosen semantics. In the OBDA (Ontology-Based Data Access) setting, many inconsistency-tolerant semantics have been proposed to query inconsistent data (Lembo et al. (2010), Bienvenu (2012)). They are generally based on the notion of data repairs, i.e. subsets of maximally consistent data. It has been recently proved that sceptically acceptance under preferred or stable semantics in argumentation theory and ODBA ICR-entailment (Intersection of Closed Repairs) are equivalent (Croitoru and Vesic (2013)). In this setting, we can consider the following Query Failure explanation problem: given an inconsistent knowledge base and an identity link, why this identity link is not entailed by the knowledge base ? When a sameAs link is not entailed under the ICR semantics, an explanation of the reasons against this entailment could be provided (including details on facts, rules and negative constraints). This dialogue would support a domain expert to detect inconsistencies and eventually correct erroneous data, revise some of the logical rules used for the invalidation or even decide the redesign of a linking strategy.

4.2.2 Weak identity links and abstract/multiscale objects

When two resources are not the same they are not necessarily completely different. For example, it is possible that erroneous identity links actually involve resources which in some way represent the same abstract concept but at different levels of details. Let b1 and b2 be two book editions that are entitled *Le roi des Aulnes* written by Michel Tournier. If these two books do not share the same ISBN, the same editor, the number of pages and the language, the link that should be asserted between b1 and b2 should not be a owl:sameAs. Yet, in this example, they refer to the same art of work.

Ontologies do not necessarily include semantically linked concepts representing all the different abstraction degrees. In such cases, different types of 'sameAs domain-dependent link' could be defined that correspond to different semantics. In particular, for such links, the substitution principle can only be applied for a subset of the properties of the considered class.

On the contrary, in some domain ontologies, different classes have been defined at different levels of abstraction. For example, in the setting of the Qualinca ANR project, the ABES partner (Agence Bibliographique de l'Enseignement Supérieur) uses the $FRBR_{00}$ ontology to represent their data. $FRBR_{00}$ (Functional Requirements for Bibliographie Records) is an ontology that has been developed to integrate library objects. In this ontology, differ-

ent abstraction levels of intellectual products are represented through the classes *Work*, *Expression*, *Manifestation*, *Item*. A work is defined as *a distinct intellectual or artistic creation*. An expression is defined as *the intellectual or artistic realization of a work in the form of alpha-numeric, musical, or choreographic notation, sound, image, object, movement*. A Manifestation is defined as *the physical embodiment of an expression of a work . . . all the physical objects that bear the same characteristics* while an item is *a single exemplar of a manifestation* [2] ! In such a context, the semantic links that can be discovered between these intellectual products can relie on the defined classes and rules are needed to link instances that are typed by the same class or by classes that are defined at different level of abstraction (e.g. is a book a new edition of an existing book or an edition of a new book ?).

We encounter a similar but more complex problem with data describing results of scientific experiments in the setting of the LIONES project. LIONES aims to model semantic LInks between ONtological multi-scalES objects involved in a transformation process. The application domain is the stabilization of micro-organisms, yeasts and bacteria. LIONES is an interdisciplinary project involving computer scientists of AgroParisTech and Paris Sud University and INRA experts in the biological domain. This project is funded by the Center for Data Science of IDEX Paris-Saclay. The need of concentrated micro-organisms (called starters) stabilized and in ready-to-use form increases. The control of their production process therefore becomes an important issue. This production process relies on a complex system, involving several unit operations: fermentation, cooling, concentration, formulation, freezing or lyophilisation and the storage of the stabilized micro-organism. Many data have been generated on micro-organisms at different stages of the production process by the INRA researchers. Observations that are realized during the researchers' experiments have been done at different scales: molecular scale, cells, cell populations, or mixtures made of different products. The AgroParisTech researchers have developped an ontological representation of the transformation process of micro-organisms: representation of the operation units, called steps of the process, the steps' successions, the involved objects, their observations at different scales and their changes during the process. This ontological representation need to be enriched to allow the representation and the detection of semantic links between inter-scale objects, intra-scale objects and during their changes through the different process steps. Then, the semantic links between the different objects generated by the production process need to be generated in order to be able to automatically discover domain-dependent rules that can be shown to experts. In this context, it cannot be said that molecules, cells, cell populations or mixtures that can be studied refers to the same object but at different abstraction degrees. The set of properties that is used to describe a cell can be very different from the properties used to describe the molecular level. However, objects that are observed at different scales must be linked if we want to discover that a property value observed at a given scale is correlated to another property value observed at a different scale.

In close collaboration with INRA and AgroParisTech researchers, and with Joe Raad (co-supervised PhD student), we will study different types of

[2] More informations on $FRBR_{00}$ can be found at http://www.cidoc-crm.org/docs/frbr_oo/frbr_docs/FRBRoo_V2.1_2015February.pdf

weak identity links that can be expressed between class instances in different application settings. Then, we will propose an ontology-based approach in which domain experts can specify the semantics of the needed weak identity links and methods that can automatically detect these semantic links will be defined.

4.2.3 Exploiting weak identity links to discover rules

In the setting of the LIONES project, the final aim is to help experts to reduce the environment impact of the production process while preserving some product properties. More precisely, we are interested in discovering rules that can help experts to understand correlations between observed attribute values at different scales and for different steps of a transformation process. Only some rules that instanciate specific patterns seem particularly relevant. However, these rules may be complex: they may involve many atoms, (weak) identity links, constants and even simple functions defined on numerical values.

In the Inductive Logic Programming field, many approaches have been developped to learn rules in a context where the closed world assumption can be assumed. (Galárraga et al. (2013; 2015)) have shown that it was possible to learn Horn rules from RDF datasets that contain millions of triples. To deal with the Open World Assumption, the authors have defined a confidence measure that exploits a Partial Completeness Assumption (i.e. if a property is valued for one instance, all the other values are supposed to be false for this instance). The authors have also shown that the allowed number of atoms of the rules and the possibility to discover rules with constants significantly affect the execution time (in their experiments, the maximum number of predicates that have been considered is 4). In a relational setting, some approaches exploit declarative constraints that limit the search space to syntactically defined subsets of rules (King et al. (2001)).

It is in this context that we will collaborate with INRA and AgroParisTech researchers to first define rule-patterns that are of interest for the experts. Then, we will study if existing approaches can be adapted to propose an approach that relies on the defined patterns to discover rules that are relevant for INRA researchers in the biological domain.

Bibliography

Mustafa Al-Bakri, Manuel Atencia, Steffen Lalande, and Marie-Christine Rousset. Inferring same-as facts from linked data: An iterative import-by-query approach. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 9–15, 2015. (Cité page 48.)

Arvind Arasu and Hector Garcia-Molina. Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 337–348. ACM Press, 2003. ISBN 1-58113-634-X. (Cité pages 8 et 9.)

Arvind Arasu, Christopher Ré, and Dan Suciu. Large-scale deduplication with constraints using dedupalog. In *ICDE*, pages 952–963, 2009. (Cité page 48.)

Manuel Atencia, Michel Chein, Madalina Croitoru, Jérôme David, Michel Leclère, Nathalie Pernelle, Fatiha Saïs, François Scharffe, and Danai Symeonidou. Defining key semantics for the RDF datasets: Experiments and evaluations. In *Graph-Based Representation and Reasoning - 21st International Conference on Conceptual Structures, ICCS 2014, Iaşi, Romania, July 27-30, 2014, Proceedings*, pages 65–78, 2014. (Cité page 48.)

Manuel Atencia, Jérôme David, and François Scharffe. Keys and pseudo-keys detection for web datasets cleansing and interlinking. In *EKAW*, pages 144–153, 2012. (Cité pages 48, 57, 68 et 69.)

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, pages 722–735, 2007. (Cité pages 3 et 8.)

Nathalie Aussenac-Gilles and Marie-Paule Jacques. Designing and evaluating patterns for ontology enrichment from texts. In *EKAW*, pages 158–165, 2006. (Cité page 17.)

Rohan Baxter, Peter Christen, and Tim Churches. A comparison of fast blocking methods for record linkage. In *ACM workshop on Data cleaning Record Linkage and Object identification*, 2003. (Cité page 36.)

Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. Tabel: Entity linking in web tables. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, pages 425–441, 2015. (Cité page 8.)

Ravish Bhagdev, Sam Chapman, Fabio Ciravegna, Vitaveska Lanfranchi, and Daniela Petrelli. Hybrid search: Effectively combining keywords and semantic searches. In *The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings*, pages 554–568, 2008. (Cité page 17.)

Meghyn Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012. (Cité page 72.)

Mustafa Bilgic, Louis Licamele, Lise Getoor, and Ben Shneiderman. Graph drawing: 13th international symposium, gd 2005, limerick, ireland, september 12-14, 2005. revised papers. Chapitre D-Dupe: An Interactive Tool for Entity Resolution in Social Networks, pages 505–507. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-31667-1. (Cité page 36.)

Alexander Bilke and Felix Naumann. Schema matching using duplicates. In *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*, pages 69–80, 2005. (Cité page 36.)

P. Borislav, K. Atanas, K. Angel, M. Dimitar, O. Damyan, and G. Miroslav. Kim - semantic annotation platform. 10(3-4):375–392, 2004. (Cité page 8.)

P. Brézellec and H. Soldano. Tabata: a learning algorithm performing a bidirectional search in a reduced search space using a tabu stratégie. In H. Prade, éditeur, *Proceedings of the 13th European Conference on Artificial Intelligence*, pages 420–424, Brighton, Angleterre, 1998. J. Wiley. (Cité page 4.)

Patrice Buche, Juliette Dibie-Barthélemy, Ollivier Haemmerlé, and Mounir Houhou. Towards flexible querying of xml imprecise data in a dataware house opened on the web. In *Flexible Query Answering Systems (FQAS)*. Springer Verlag, june 2004. (Cité pages 5 et 9.)

Patrice Buche, Juliette Dibie-Barthélemy, Liliana Ibanescu, and Lydie Soler. Fuzzy web data tables integration guided by an ontological and terminological resource. *IEEE Trans. Knowl. Data Eng.*, 25(4):805–819, 2013. (Cité page 16.)

Paul Buitelaar and Melanie Siegel. Ontology-based information extraction with soba. In *Proc. of the International Conference on Language Resources and Evaluation (LREC*, pages 2321–2324, 2006. (Cité page 17.)

Michael J. Cafarella, Alon Y. Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. Webtables: exploring the power of tables on the web. *PVLDB*, 1(1): 538–549, 2008. (Cité pages 5 et 8.)

Pablo Castells, Miriam Fernández, and David Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Trans. Knowl. Data Eng.*, 19(2):261–272, 2007. (Cité page 17.)

Chin-Liang Chang and Richard Char-Tung Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, Inc., Orlando, FL, USA, 1997. ISBN 0121703509. (Cité page 39.)

Paolo Ciccarese, Marco Ocana, and Tim Clark. Open semantic annotation of scientific publications using DOMEO. *J. Biomedical Semantics*, 3(S-1), 2012. (Cité page 7.)

Philipp Cimiano, Siegfried Handschuh, and Steffen Staab. Towards the self-annotating web. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 462–471, New York, NY, USA, 2004. ACM. ISBN 1-58113-844-X. (Cité page 13.)

Philipp Cimiano, Gunter Ladwig, and Steffen Staab. Gimme'the context : Context driven automatic semantic annotation with c-pankow. In *WWW conference*, 2005. (Cité pages 8 et 18.)

William W. Cohen. Data integration using similarity joins and a word-based information representation language. *ACM Transactions on Information Systems*, 18(3):288–321, 2000. (Cité page 36.)

William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, pages 73–78, 2003. (Cité pages 35 et 42.)

William W. Cohen and Jacob Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *KDD '02*, pages 475–480, NY, USA, 2002. ACM. ISBN 1-58113-567-X. (Cité page 44.)

Olivier Corby, Rose Dieng-Kuntz, Catherine Faron-Zucker, and Fabien L. Gandon. Searching the semantic web: Approximate query processing based on ontologies. *IEEE Intelligent Systems*, 21(1):20–27, 2006. (Cité page 17.)

Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Automatic web information extraction in the roadrunner system. In *Revised Papers from the HUMACS, DASWIS, ECOMO, and DAMA on ER 2001 Workshops*, pages 264–277. Springer-Verlag, 2002. ISBN 3-540-44122-0. (Cité pages 8 et 9.)

Madalina Croitoru and Kees van Deemter. A conceptual graph approach for the generation of referring expressions. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2456–2461, 2007. (Cité page 70.)

Madalina Croitoru and Srdjan Vesic. What can argumentation do for inconsistent ontology query answering? In *Scalable Uncertainty Management - 7th International Conference, SUM 2013, Washington, DC, USA, September 16-18, 2013. Proceedings*, pages 15–29, 2013. (Cité page 72.)

Robert Dale and Ehud Reiter. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive science*, 19(2): 233–263, 1995. (Cité page 70.)

G. de Melo. Not quite the same: Identity constraints for the Web of Linked Data. In *Proc. of the 27th Conference on Artificial Intelligence*. AAAI Press, 2013. (Cité pages 60 et 71.)

AnHai Doan, Ying Lu, Yoonkyong Lee, and Jiawei Han. Profile-based object matching for information integration. *Intelligent Systems, IEEE*, 18(5):54– 59, September/October 2003. (Cité page 10.)

Xin Dong, Alon Halevy, and Jayant Madhavan. Reference reconciliation in complex information spaces. In *ACM SIGMOD*, pages 85–96. ACM Press, 2005. ISBN 1595930604. (Cité pages 6, 36 et 44.)

P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and< i> n</i>-person games. *Artificial intelligence*, 77(2):321–357, 1995. (Cité page 72.)

Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19:1–16, 2007. ISSN 1041-4347. (Cité pages 34 et 36.)

Ivan P. Fellegi and Alan B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969. (Cité page 34.)

Alfio Ferrara, Andriy Nikolov, and François Scharffe. Data linking for the semantic web. *Int. J. Semantic Web Inf. Syst.*, 7(3):46–76, 2011. (Cité page 36.)

Alfio Ferrara, Andriy Nikolov, and François Scharffe. Data linking. *J. Web Sem.*, 23:1, 2013. (Cité pages 2 et 6.)

Kenneth D. Forbus and Johan de Kleer. *Building problem solvers*. MIT Press, Cambridge, MA, USA, 1993. ISBN 0-262-06157-0. (Cité page 39.)

Hélène Gagliardi, Ollivier Haemmerlé, Nathalie Pernelle, and Fatiha Saïs. An automatic ontology-based approach to enrich tables semantically. In *In proceedings of AAAI-2005 C&O workshop*, 2005a. (Cité pages 5 et 9.)

Hélène Gagliardi, Ollivier Haemmerlé, Nathalie Pernelle, and Fatiha Saïs. A semantic enrichment of data tables applied to food risk assessment. In *Discovery Science, 8th International Conference, DS 2005, Singapore, October 8-11, 2005, Proceedings*, pages 374–376, 2005b. (Cité pages 5 et 9.)

Souhir Gahbiche, Nathalie Pernelle, and Fatiha Saïs. Explaining reference reconciliation decisions: A coloured petri nets based approach. In *Advances in Knowledge Discovery and Management - Volume 2 [Best of EGC 2010, Hammamet, Tunisie]*, pages 63–81, 2010a. (Cité pages 6 et 35.)

Souhir Gahbiche, Nathalie Pernelle, and Fatiha Saïs. Explication de décisions de réconciliation de références : approche fondée sur les réseaux de petri colorés. In *Extraction et gestion des connaissances (EGC'2010), Actes, 26 au 29 janvier 2010, Hammamet, Tunisie*, pages 327–338, 2010b. (Cité page 35.)

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. Fast rule mining in ontological knowledge bases with AMIE+. *VLDB Journal*, 24 (6):707–730, 2015. (Cité pages 71 et 74.)

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 413–422, 2013. (Cité pages 71 et 74.)

D. Gerber and A.-C. Ngonga Ngomo. Bootstrapping the linked data web. In *1st Workshop on Web Scale Knowledge Extraction, International Semantic Web Conference (1)*, volume 7031 de *Lecture Notes in Computer Science*. Springer, 2011. ISBN 978-3-642-25072-9. (Cité page 30.)

R. Godin, G.W. Mineau, R. Missaoui, and H. Mili. Méthodes de classification conceptuelle basées sur les treillis de galois et applicationz. *Revue d'Intelligence Artificielle*, 9(2):105–137, 1995. (Cité page 4.)

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Press, Baltimore, MD, USA, second édition, 1989. (Cité page 43.)

Dimitrios Gunopulos, Roni Khardon, Heikki Mannila, Sanjeev Saluja, Hannu Toivonen, and Ram Sewak Sharma. Discovering all most specific sentences. *ACM Trans. Database Syst.*, 28(2):140–174, 2003. (Cité page 68.)

H. Halpin and P.J. Hayes. When owl: sameas isn't the same: An analysis of identity links on the semantic web. In *Proc. of the WWW2010 Workshop on Linked Data on the Web*, 2010. (Cité page 71.)

H. Halpin, P.J. Hayes, and H.S. Thompson. When owl: sameas isn't the same redux: A preliminary theory of identity and inference on the semantic web. In *Workshop on Discov. Meaning On the Go in Large Heterogeneous Data (LHD-11), Barcelona, Spain, July 16, 2011*, pages 25–30, 2011. (Cité pages 60 et 71.)

J. Heflin and H. Muñoz-avila. LCW-based agent planning for the semantic web. In *Ontologies and the Semantic Web Workshop*, pages 63–70. AAAI Press, 2002. (Cité page 63.)

Jeff Heflin, James A. Hendler, and Sean Luke. SHOE: A blueprint for the semantic web. In *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential [outcome of a Dagstuhl seminar]*, pages 29–63, 2003. (Cité page 7.)

Lawrence J. Henschen and Larry Wos. Unit refutations and horn sets. *J. ACM*, 21(4):590–605, 1974. (Cité page 39.)

Joachim Hereth, Gerd Stumme, Rudolf Wille, and Uta Wille. Conceptual knowledge discovery and data analysis. In *International Conference on Conceptual Structures*, pages 421–437, 2000. (Cité page 4.)

Gaëlle Hignette, Patrice Buche, Juliette Dibie-Barthélemy, and Ollivier Haemmerlé. Fuzzy annotation of web data tables driven by a domain ontology. In *ESWC*, pages 638–653, 2009. (Cité page 17.)

Chun-Nan Hsu and Ming-Tzung Dung. Generating finite-state transducers for semi-structured data extraction from the web. *Inf. Syst.*, 23(9):521–538, 1998. (Cité page 9.)

Wei Hu, Jianfeng Chen, and Yuzhong Qu. A self-training approach for resolving object coreference on the semantic web. In *WWW*, pages 87–96, 2011. (Cité pages 36, 48 et 70.)

Carlos A. Hurtado, Alexandra Poulovassilis, and Peter T. Wood. A relaxed approach to RDF querying. In *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, pages 314–328, 2006. (Cité page 17.)

R.J. Hyndman and Y. Fan. Sample quantiles in statistical packages. *The American Statistician*, 50:361–365, 1996. (Cité page 69.)

Robert Isele and Christian Bizer. Learning expressive linkage rules using genetic programming. *PVLDB*, 5(11):1638–1649, 2012. (Cité page 48.)

Kurt Jensen. *Coloured Petri Nets, Basic Concepts*. Springer, 1997. (Cité page 45.)

Ross D. King, Ashwin Srinivasan, and Luc Dehaspe. Warmr: a data mining tool for chemical data. *Journal of Computer-Aided Molecular Design*, 15(2): 173–181, 2001. (Cité page 74.)

Hanna Köpcke and Erhard Rahm. Frameworks for entity matching: A comparison. *Data Knowl. Eng.*, 69(2):197–210, 2010. (Cité page 34.)

Nicholas Kushmerick. Wrapper induction: efficiency and expressiveness. *Artif. Intell.*, 118(1-2):15–68, 2000. ISSN 0004-3702. (Cité page 9.)

Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In *Web Reasoning and Rule Systems - Fourth International Conference, RR 2010, Bressanone/Brixen, Italy, September 22-24, 2010. Proceedings*, pages 103–117, 2010. (Cité page 72.)

Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. Rimom: A dynamic multistrategy ontology alignment framework. *IEEE Trans. Knowl. Data Eng.*, 21(8):1218–1232, 2009. (Cité page 36.)

Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. Annotating and searching web tables using entities, types and relationships. *PVLDB*, 3(1): 1338–1347, 2010. (Cité pages 5, 8 et 17.)

Wai Lup Low, Mong Li Lee, and Tok Wang Ling. A knowledge-based approach for duplicate elimination in data cleaning. *Information Systems*, 26:585–606, December 2001. ISSN 0306-4379. (Cité page 48.)

Pablo N. Mendes, Max Jakob, Andres Garcia-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, 2011. (Cité page 7.)

Yassine Mrabet, Nacéra Bennacer, and Nathalie Pernelle. Controlled knowledge base enrichment from web documents. In *Web Information Systems Engineering - WISE 2012 - 13th International Conference, Paphos, Cyprus, November 28-30, 2012. Proceedings*, pages 312–325, 2012. (Cité pages 5 et 17.)

Yassine Mrabet, Nacéra Bennacer, and Nathalie Pernelle. Controlled knowledge base enrichment from web documents. *Revue des Sciences et Technologies de l'Information - Série RIA : Revue d'Intelligence Artificielle*, 28(2-3):297–320, 2014. (Cité pages 5 et 17.)

Yassine Mrabet, Nacéra Bennacer, Nathalie Pernelle, and Mouhamadou Thiam. Supporting semantic search on heterogeneous semi-structured documents. In *Advanced Information Systems Engineering, 22nd International Conference, CAiSE 2010, Hammamet, Tunisia, June 7-9, 2010. Proceedings*, pages 224–229, 2010. (Cité pages 5, 17 et 18.)

Ion Muslea, Steven Minton, and Craig A. Knoblock. Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems*, 4(1-2):93–114, 2001. ISSN 1387-2532. (Cité page 9.)

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. pages 3–26, 2007. (Cité page 7.)

H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P. James. Automatic linkage of vital records. *Science*, 130:954–959, October 1959. ISSN 0036-8075. (Cité page 34.)

Howard B. Newcombe and James M. Kennedy. Record linkage: making maximum use of the discriminating power of identifying information. *Commun. ACM*, 5(11):563–566, 1962. ISSN 0001-0782. (Cité page 36.)

Axel-Cyrille Ngonga Ngomo and Klaus Lyko. Eagle: Efficient active learning of link specifications using genetic programming. In *9th Extended Semantic Web Conference (ESWC)*, pages 149–163, 2012. (Cité page 48.)

Andriy Nikolov, Mathieu d'Aquin, and Enrico Motta. Unsupervised learning of link discovery configuration. In *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*, pages 119–133, 2012a. (Cité page 36.)

Andriy Nikolov, Mathieu d'Aquin, and Enrico Motta. Unsupervised learning of link discovery configuration. In *9th Extended Semantic Web Conference (ESWC)*, pages 119–133, Berlin, Heidelberg, 2012b. Springer-Verlag. ISBN 978-3-642-30283-1. (Cité page 48.)

Laura Papaleo, Nathalie Pernelle, Fatiha Saïs, and Cyril Dumont. Logical detection of invalid sameas statements in RDF data. In *Knowledge Engineering and Knowledge Management - 19th International Conference, EKAW 2014, Linköping, Sweden, November 24-28, 2014. Proceedings*, pages 373–384, 2014. (Cité pages 6, 60 et 71.)

Singla Parag and Domingos Pedro. Multi-relational record linkage. In *MRDM Workshop*, 2004. (Cité page 44.)

Nathalie Pernelle, Marie-Christine Rousset, Henry Soldano, and Véronique Ventos. Zoom: a nested galois lattices-based system for conceptual clustering. *Journal of Experimentical and Theoritical. Artificial Intelligence*, 14(2-3): 157–187, 2002. (Cité page 4.)

Nathalie Pernelle, Marie-Christine Rousset, and Véronique Ventos. Automatic construction and refinement of a class hierarchy over multi-valued data. In *Principles of Data Mining and Knowledge Discovery, 5th European Conference, PKDD 2001, Freiburg, Germany, September 3-5, 2001, Proceedings*, pages 386–398, 2001. (Cité page 3.)

Nathalie Pernelle and Fatiha Saïs. Classification rule learning for data linking. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops, Berlin, Germany, March 30, 2012*, pages 136–139, 2012. (Cité page 35.)

Nathalie Pernelle, Fatiha Saïs, Brigitte Safar, Maria Koutraki, and Tushar Ghosh. N2r-part: identity link discovery using partially aligned ontologies. In *Proceedings of the 2nd International Workshop on Open Data, WOD 2013, Paris, France, June 3, 2013*, pages 6:1–6:4, 2013a. (Cité page 35.)

Nathalie Pernelle, Fatiha Saïs, and Danai Symeonidou. An automatic key discovery approach for data linking. *Journal of Web Sem.*, 23:16–30, 2013b. (Cité pages 6, 47, 57 et 68.)

Nathalie Pernelle, Fatiha Saïs, and Danai Symeonidou. An automatic key discovery approach for data linking. *J. Web Sem.*, 23:16–30, 2013c. (Cité page 47.)

Nathalie Pernelle, Danai Symeonidou, and Fatiha Saïs. Une approche de découverte de clés conditionnelles dans des données rdf. In *Proceedings of the french conference Ingénierie des Connaissances*, pages 231–236, 2015. (Cité page 47.)

Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001. ISSN 1066-8888. (Cité page 10.)

Yuan Ren, Kees van Deemter, and Jeff Z. Pan. Generating referring expressions with OWL2. In *Proceedings of the 23rd International Workshop on Description Logics (DL 2010), Waterloo, Ontario, Canada, May 4-7, 2010*, 2010. (Cité page 70.)

Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal ACM*, 12(1):23–41, 1965. ISSN 0004-5411. (Cité page 38.)

Yannis Roussakis, Ioannis Chrysakis, Kostas Stefanidis, Giorgos Flouris, and Yannis Stavrakas. A flexible framework for understanding the dynamics of evolving RDF datasets. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, pages 495–512, 2015. (Cité page 68.)

Fatiha Saïs, Hélène Gagliardi, Ollivier Haemmerlé, and Nathalie Pernelle. Enrichissement sémantique de documents XML représentant des tableaux. In *Extraction et gestion des connaissances (EGC'2005), Actes des cinquièmes journées Extraction et Gestion des Connaissances, Paris, France, 18-21 janvier 2005, 2 Volumes*, pages 407–418, 2005. (Cité page 9.)

Fatiha Saïs, Nobal B. Niraula, Nathalie Pernelle, and Marie-Christine Rousset. Ln2r a knowledge based reference reconciliation system: Oaei 2010 results. In *Proceedings of the 5th International Workshop on Ontology Matching (OM-2010)*, 2010. (Cité pages 6, 35 et 44.)

Fatiha Saïs, Nathalie Pernelle, and Marie-Christine Rousset. L2R: A Logical Method for Reference Reconciliation. In *AAAI*, pages 329–334, 2007. (Cité pages 6 et 35.)

Fatiha Saïs, Nathalie Pernelle, and Marie-Christine Rousset. Combining a logical and a numerical method for data reconciliation. *Journal on Data Semantics*, 12:66–94, 2009. (Cité pages 6, 35, 48, 63 et 64.)

Sunita Sarawagi and Alok Kirpal. Efficient set joins on similarity predicates. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, pages 743–754, New York, NY, USA, 2004. ACM. ISBN 1-58113-859-8. (Cité page 35.)

Pavel Shvaiko and Jérôme Euzenat. Ontology matching: State of the art and future challenges. *IEEE Trans. Knowl. Data Eng.*, 25(1):158–176, 2013. (Cité pages 2 et 34.)

Yannis Sismanis, Paul Brown, Peter J. Haas, and Berthold Reinwald. Gordian: efficient and scalable discovery of composite keys. In *Proceedings of the 32nd International conference Very Large Data Bases (VLDB)*, VLDB '06, pages 691–702. VLDB Endowment, 2006. (Cité pages 51 et 52.)

D. Song and J. Heflin. Automatically generating data linkages using a domain-independent candidate selection approach. In *International Semantic Web Conference*, pages 649–664, 2011. (Cité page 36.)

Tommaso Soru, Edgard Marx, and Axel-Cyrille Ngonga Ngomo. ROCKER: A refinement operator for key discovery. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1025–1033, 2015. (Cité pages 48, 57 et 68.)

Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. PARIS: probabilistic alignment of relations, instances, and schema. *PVLDB*, 5(3):157–168, 2011. (Cité page 36.)

Fabian M. Suchanek, Georgiana Ifrim, and Gerhard Weikum. Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, page 712, New York, New York, USA, Août 2006. ISBN 1595933395. (Cité pages 8 et 17.)

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 697–706, 2007. (Cité pages 3 et 8.)

Fabian M. Suchanek, Mauro Sozio, and Gerhard Weikum. SOFIE: a self-organizing framework for information extraction. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 631–640, 2009. (Cité pages 8 et 17.)

Danai Symeonidou, Vincent Armant, Nathalie Pernelle, and Fatiha Saïs. Sakey: Scalable almost key discovery in RDF data. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, pages 33–49, 2014. (Cité pages 6, 47, 57, 63, 64, 68 et 69.)

Danai Symeonidou, Nathalie Pernelle, and Fatiha Saïs. Kd2r: A key discovery method for semantic reference reconciliation. In *OTM Workshops*, pages 392–401, 2011. (Cité pages 6 et 47.)

Mouhamadou Thiam, Nacéra Bennacer, Nathalie Pernelle, and Moussa Lo. Incremental ontology-based extraction and alignment in semi-structured documents. In *Database and Expert Systems Applications, 20th International Conference, DEXA 2009, Linz, Austria, August 31 - September 4, 2009. Proceedings*, pages 611–618, 2009. (Cité pages 17 et 24.)

Mouhamadou Thiam, Nathalie Pernelle, and Nacéra Bennacer. Contextual and metadata-based approach for the semantic annotation of heterogeneous documents. In *First International Workshop on Semantic Metadata Management and Applications, SeMMA 2008, Located at the Fifth European Semantic Web Conference (ESWC 2008), Tenerife, Spain, June 2nd, 2008. Proceedings*, pages 18–30, 2008. (Cité page 17.)

Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and maintaining links on the web of data. In *Proceedings of the 8th International Semantic Web Conference*, ISWC '09, pages 650–665, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-04929-3. (Cité pages 36 et 48.)

William E Winkler. Overview of record linkage and current research directions. Rapport technique, Statistical Research Division U.S. Census Bureau Washington, DC 20233, 2006a. (Cité pages 6 et 34.)

William E. Winkler. Overview of record linkage and current research directions. Rapport technique, Bureau of the Census, 2006b. (Cité page 36.)

Mohamed Amir Yosef, Johannes Hoffart, Ilaria Bordino, Marc Spaniol, and Gerhard Weikum. AIDA: an online tool for accurate disambiguation of named entities in text and tables. *PVLDB*, 4(12):1450–1453, 2011. (Cité page 7.)

JM R. Yves, E.P. Shironoshita, and M.R. Kabuka. Ontology matching with semantic verification. *Web Semantics*, 7(3):235–251, Septembre 2009. ISSN 1570-8268. (Cité pages 63 et 64.)

Fouad Zablith, Grigoris Antoniou, Mathieu d'Aquin, Giorgos Flouris, Haridimos Kondylakis, Enrico Motta, Dimitris Plexousakis, and Marta Sabou. Ontology evolution: a process-centric survey. *The Knowledge Engineering Review*, 30:45–75, 1 2015. ISSN 1469-8005. (Cité page 68.)

Ziqi Zhang. Learning with partial data for semantic table interpretation. In *Knowledge Engineering and Knowledge Management - 19th International Conference, EKAW 2014, Linköping, Sweden, November 24-28, 2014. Proceedings*, pages 607–618, 2014. (Cité pages 5 et 8.)