PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

Knowledge-Based Open Entity Matching

Stefano Bortoli

Advisor:

Prof. Paolo Bouquet

Università degli Studi di Trento

April 2013

Preface

The path that lead me to the realization of this work starts back in early 2005. At that time I had just completed my bachelor degree in Computer Science at the University of Trento, and started working as research assistant on the EU funded VIKEF project in the group of professor Bouquet. Working on that project, we had to deal for the first time with problems related to the integration of large RDF graphs resulting from diverse automatic data extraction processes. Quickly we realized that a viable solution to the problem would have been to assign a priori the same URI to the resources we wanted to mention unambiguously in the different RDF graphs. This intuition follows the principles of the Occam's Razor, suggesting to avoid the unnecessary multiplications of identifiers for the same entity. Thereby, under the coordination of professor Bouquet we started to conceive and implement the first prototype of what we called Entity Repository. The discussions and brainstorming necessary to the conception and first implementation of the prototype fostered the development of a more ambitious vision: define a global naming service for the creation and maintenance of globally unique identifiers for non-web resources. Following this objective, professor Bouquet worked to form a consortium of prestigious research institutes and companies, and successfully obtained a consistent EU grant to develop the idea. After three years of intense and challenging work, in 2009 the OKKAM project consortium released a first complete working prototype of the Entity Name System (ENS). The ENS embeds in a scalable architecture state of the art solutions in many fields of information science. The ENS was considered a real success with plenty of potential applications in the real world cases. Therefore, with the support of the closest partners, professor Bouquet founded a spin-off company to sustain and further develop the vision of a Semantic Web where frictionless entity-centric information integration was possible. Nevertheless, despite the efforts of brilliant researches, the solutions defined to the most challenging problem of entity matching required further specialization and development. In this context, I decided to apply for a PhD scholarship at the International Doctoral School in ICT of the University of Trento, and start my research to define a more effective and efficient solution to the entity matching problem under the supervision of professor Bouquet. Needless to say that since then, the path to the definition this thesis passed through tons of papers, a fruitful visit at the Information Sciences Institute in Los Angeles, moments of excitements due to discovery, and depressive moments due to failures in the evaluation of intuitions. Most of all, what characterized this period was a constant struggling between the will of following interesting leads in the solutions of possibly marginal aspects of the problem and the need of staying focused on the target. I am not entirely sure the work proposed in this thesis is the best possible I could have done. Probably, if I could do it again, I would do it differently. For sure, I did not save energies, passion and commitment. The adrenaline, happiness and satisfaction coming from the successful testing of a new solution always compensates for long working hours, and weekends and holidays spent reading and programming. Sometimes one may even get lost, but I believe that getting lost is a necessary condition to push oneself in the research of innovative solutions. One of my professors in high school used to tell to the students: "when you don't understand anything about something, it is the moment you start learning about it". Well, learning is what I have done, and learning is what I want to do. I sincerely believe that what is presented in this work is neither revolutionary, nor conclusive. At the same time, I am convinced that this work in its broadness is the cornerstone for the development of innovative solutions that contribute to move a step ahead towards the realization of the vision I embraced back in 2005.

Contents

Chapter 1

Introduction

In the very promising vision of the Semantic Web, software agents are capable of exploiting semantically annotated information in order to perform automatically time expensive tasks on behalf of human users [12, 2]. In order to realize this ambitious goal, much effort has been spent in the past to define and spread the usage of tools aiding the production of semantically annotated information. As a result, software agents are expected to be able of gathering, exchanging, automatically processing and integrating semantically structured information to perform sophisticated tasks. Nevertheless, there are still many subtle and complex philosophical issues to be solved, undermining the establishment of a solid foundation of the whole architecture of the (Semantic) Web. One of the debated points is related to the problem of identity and reference on the Web [75, 25, 24, 11, 94, 44]. Namely, there is a disagreement about the way syntactically unique identifiers (i.e. Uniform Resource Identifiers) should be tied to resources (i.e. entities) in order to provide unambiguous means of reference (i.e. names) and how these should fit the current architecture of the Web. In the first concrete realization of the Semantic Web vision known as Linked Data [13, 20], the proliferation of identifiers for entities is deliberately allowed, relying on the assumption that, with time, conventions will emerge. Notice that in the real world, the society deals with the identification issue by convention, e.g. passport, social security number, ISBN, computer network card MAC address, Web Domain names, etc. However, despite some concrete attempts, e.g. [25], the community currently contributing to the development of the Semantic Web seems to be reluctant to adopt any policy of naming conventions; these are perceived as authoritarian and against the free nature of the Web. Still, the promising vision of a world where software agents, exploring semantically structured knowledge, are able to perform advanced and critical tasks is exciting, even when no naming convention is commonly shared. In order to automatically integrate semantically structured information outside of naming conventions, software agents ought to deal with several complex, multi-faceted issues. One of them is related to the solution of the entity matching problem, which can be particularly complicated considering the heterogeneous and ambiguous knowledge available on the Web [75]. In a few words, the entity matching problem consists in establishing whether two entity descriptions (i.e. sets of attributes) refer to the same real world entity or not.

The entity matching problem is well known in the context of information systems and databases. In fact, for many different reasons it is often necessary to seek for duplicate records in large databases. Several effective solutions to this problem have been proposed and applied. Most of these solutions were conceived to be adopted in a controlled environment, under the supervision of experts responsible for taking educated decisions when automatic methods required clerical review. Considering the *una tantum* nature of the operation executed in a closed controlled environment, these types of solutions were, and are acceptable. However, if we consider the entity matching problem in the context of the Web, it is easy to realize that many of the assumptions underlying satisfactory entity matching solutions are no longer effective. For example, the domain of interpretation is too broad to be mastered by an expert, and the execution of traditional Export, Transformation, and Loading (ETL) operations becomes particularly complicated due to inherent semantic and structural heterogeneity that characterizes the data available on the Web. Moreover, given the scale and amount of data available, it is impossible to apply many heuristic techniques that lead to effective solutions in a limited, controlled environment (e.g. threshold optimization).

Most of the existing solutions to the entity matching problem - including the most famous Theory of Record Linkage of Fellegi and Sunter [60] - rely on the definition of probabilistic and similarity thresholds that support entity matching decisions. In a closed context, it is plausible to assume the precise setting of such thresholds, supporting reliable matching decisions on the majority of the cases. However, this becomes particularly complicated, if not impossible to realize, in the context of the Web. Moreover, outside of the context of definition, probability and distance scores can hardly be interpreted, becoming useful solely for ranking purposes. The selection of the best score on top of a ranked list can be suitable to support positive matching decisions. However, this approach starts to show its limitations when considering a single pair of descriptions. When can we consider a similarity score high or low enough to support reliable positive or negative matching decisions? Answers to this question have been given in several works (for example [40]). However, threshold optimization is not possible in a global, open environment. In fact, to find the optimal probability likelihood (or similarity threshold) that minimizes classification error we would need complete knowledge about any entity of the world. In this work, we assume that in principle complete knowledge is not achievable. A way to overcome the threshold dilemma is to define explicit matching rules, dictating the logic of matching, as suggested in [103, 79]. The solution of entity matching relying on rules system is very convenient because in general, rules are self-explanatory, explicit, intelligible and can be used to justify and support the evaluation of entity matching decisions. The main drawbacks of rule-based systems are the considerable amount of human effort required to create and maintain rules for entity matching, and the fact that they usually define very sharp conditions that may not make them applicable in all cases. In fact, the manual definition of custom rules to match pairs of descriptions as proposed for example in SILK [147], suffers of scalability issues. As a matter of fact, if we want to solve entity matching problem between N different sources, we have to define N × N-1 SILK linkage rules, one for each pair of sources. To overcome this problem, recent works propose advanced methods for automatically mining such entity matching rules [88, 118]. This approach may reduce the human effort required to define rules, but usually these methods are not easy to employ. What's more, the rules constructed are only suitable to tackle the problem on the datasets for which the rules are built. Also in this case, the scalability of the solution is seriously reduced.

Following these observations, the present thesis aims to define a novel knowledge-based entity matching framework for the implementation of a reliable and incrementally scalable solution to the entity matching problem in the context of the Web. The founding pillars of the solution we propose in this work are (1) a lightweight ontology defining the entity types and the features relevant for the solution of the entity matching problem, and (2) a set of entity matching rules expressed in terms of the ontology forming an equational theory to support reliable entity matching decisions. The ontology, representing the types and features considered, has a two-fold role: on the one hand, it provides a base for discerning relevant features from the irrelevant ones; on the other hand, it provides a central point of reference for the definition of contextual semantic mappings which ease the problem of semantic heterogeneity. Rules defined in terms of the ontology can be applied and reused to match any descriptions pair once the semantic of the feature has been harmonized towards the defined ontology. It is important to stress that this work does not address the problem of automatic definition of ontological mappings, as many solutions already exist [59]. In this work, the analysis of this problem is limited by assuming the existence of such mappings.

The practical realization of the proposed solution includes the definition of a process to support the building and maintenance of generic and effective entity matching rules. In this regard, it is interesting to notice that when required, people are capable of solving the entity matching problem quite effectively, relying on a combination of heuristic, background knowledge, and common sense. In particular, when looking at descriptions to be matched, people are capable of isolating the features that are relevant for matching, and take accurate matching decisions. To confirm this observation, recent works have proposed to solve the entity matching problem relying also on crowd-sourcing [148, 117]. Therefore, capturing this type of *matching knowledge*, seems to be a suitable lead for the definition of reliable entity matching rules. Notice that this does not imply that people will be asked to explicitly define and maintain the rules, but rather we aim to extract rules based on people's feedback about matching decisions on ambiguous cases. This would enable us to use machine learning techniques to automatically elicit entity matching rules, and thereby, support the scalability and sustainability of the knowledge-based solution. Furthermore, recent trends related to the development of the Web 2.0 showed how the creation of a community of interest can help to scale up the human effort required.

Still, relying purely on machine learning techniques to construct entity matching rules may be limiting in that available data may not contain all the information necessary to define a complete set of rules. For example, it is known that some properties can be very effective in driving matching decisions (e.g. email address), but we can hardly find this type of information in open public data sources. Furthermore, fuzziness in the data and human errors could decrease the quality and the reliability of the elicited rules. Therefore, this work proposes to integrate rules that result from a bottom-up, machine learning process, with entity matching rules resulting from a formal ontological analysis of the features defined in the ontology. Our intuition is that the combination of top-down and bottom-up rules through a specialized merging process will lead to the definition of a more complete and effective set of entity matching rules to be employed in the context of the Web.

The definition of a generic knowledge-based entity matching process requires to approach the research for a solution in a multi-disciplinary manner, exploring the latest advancements of different communities dealing with diverse aspects of the problem. In particular, we should explore the scientific tools provided by disciplines dealing with knowledge representation and human decision processes, such as cognitive science and philosophy, to ground the "knowledge base pillar" of the overall matching process. We are aware that in principle knowledge-based solutions may be negatively affected by inconsistencies and errors of data available on the Web. In any case, we believe that it is worthy to explore the definition of such solution and to implement and evaluate its bootstrap, confident that the creation of a community of interest would render sustainable the effort necessary for the incremental definition of improvements and refinements.

The reminder of this thesis is organized as follows: in the first part, we formally define the problem (chapter 2) and we present a review of the state of the art in the solution of the entity matching problem (chapter 3); in the second part, we present in detail the vision underlying the solution we propose (chapter 4) and we formally define and describe the ontology developed (chapter 5), and the rules for entity matching (chapter 6); in the third part of the thesis, we describe the implementation of what we formally defined in the second part, providing details about the solution of the semantic and structural heterogeneity (chapter 7), describing the process of construction of rules for entity matching (chapter 8), and presenting solutions related to the practical execution of entity matching problem as a software program (chapter 9). Finally in chapter 10 we propose to validate the proposed approach through a set of validation experiments showing the impact of some of the solution proposed in this work, and then in chapter 11 we present some conclusions and future work.

1.1 Mission Statement

In this work we argue for the definition a knowledge-based entity matching framework for the implementation of a reliable and incrementally scalable solution. Such knowledge base is formed by an ontology and a set of entity matching rules suitable to be applied as a reliable equational theory in the context of the Semantic Web. In particular, we are going to prove that relying on the existence of a set of contextual mappings to ease the semantic heterogeneity characterizing descriptions on the Web, a knowledge-based solution can perform comparably, and sometimes better, than existing solutions at the state of the art.

We further argue that a knowledge-based solution to the open entity matching problem ought to be considered under the open world assumption, as in some cases the descriptions to be matched may not contain the necessary information to take any accurate matching decision.

The main goal of this work is to show how the framework proposed is suitable to pursue a reliable solution of the entity matching problem, regardless the set of rules or the ontology adopted. In fact, we believe that structural and syntactic heterogeneity affecting data on the Web undermine the definition of a global unique solution. However, we argue that a knowledge-driven approach, considering the semantic and meta-properties of compared attributes, can provide important benefits and lead to more reliable solutions. To achieve this goal, we are going to implement several experiments to evaluate different sets of rules, testing our thesis and learning important lessons for future developments. The sets of rules that we will consider to bootstrap the solution proposed in this work are the result of diverse complementary processes: first we want to investigate whether capturing the matching knowledge employed by people in taking entity matching decision by relying on machine learning techniques can produce an effective set of rules (bottom-up strategy); second, we investigate the application of formal ontology tools to analyze the features defined in the ontology and support the definition of entity matching rules (top-down strategy). Moreover, in this work we argue that by merging the rules resulting from these complementary processes, we can define a set of rules that can support reliably entity matching decision in an open context.

Part I

The Problem and the State of the Art

Chapter 2

The Problem: Matching in the Open World

The entity matching problem is known in many different fields of computer science with different names: *record linkage* or *record matching*[60], *merge purge* [80], *duplicate detection*[131], *entity* and *object identification*[95, 160] in the database community; *instance identification*[150],*database hardening*[46], *name matching*[18] among others in the Artificial Intelligence community; "*object consolidation*"[42] and "*coreference resolution*"[66] are used in the contexts of Semantic Web and Named Entity Recognition along Natural Language Processing task.

In the context of databases and information system, the problem of duplicate records creation is the result of the breakdown of the information model underlying relational databases when data stored in multiple databases ought to be integrated. In [93], Kent describes in a lucid way to problems of information integration among different databases when naming and identification are managed only considering a local (or private) scope. Ideally, a database schema tries to be present a faithful model of what we call reality. Kent asserts that the most fundamental principle of data modeling relies on the one-to-one correspondence between the proxy object in the database and entity object in the real world the proxies are supposed to represent. Naming is essential to support modeling, both referring to entities and to relations (functions). Computer programs hardly can autonomously resolve ambiguous references in specific contexts, thus they rely on unique identifying codes assigned to entities. Unfortunately, these codes can be unreliable for identifying means. Value-based systems such as relational databases don't provide good models for identity, as primary and foreign key are rough approximation. Some objects might not have primary relation in which their identifier serves as primary key; the same key might be a primary key in several tables; nothing prevents several primary keys to be assigned to the same object. This inadequate identity management requires a two step process to identify records in different database: first, identity needs to be disambiguated within each single system, and then integrated in multi-database environment. The information model starts to break down when we deal with several databases at the time and we want to present the illusion of a single database. Every creation in a database creates a new distinct proxy, so we can no longer ignore the difference between proxies and entities. Let's say $x = y$ when x and y refer to the same proxy, whereas we say $x \equiv y$ when they refer to the same entity. Notice that $x = y$ should imply $x \equiv y$, but not vice versa. Thus, creation in a database is no longer creating a new entity but it creates a new proxy referring to an entity that was not represented in a particular database. How shall we know that two proxies represent the same entity? i.e. that $x \equiv y$ even thou $x \neq y$. Hidden constraints about identifiers that were implicit in a single database emerge and create lot of problems in multi-database system. The break down of the information model underlying relational database described by Kent is the main cause of ambiguities that are often the main source of duplicate generation that requires entity matching resolution.

In a world where data persistence was managed mostly in relational databases, the solution of entity matching problem focused on database records. However, the fast growth in the amount of digital data production driven Web 2.0, cloud services and Social Web imposed a paradigm shift in the management data representation at persistence level. In fact, highly scalable NOSQL database systems underlie most of the popular web application and cloud-based solutions (e.g. Google Bigtable [38], Facebook Cassandra[1], Amazon SimpleDB[2]) providing reliable contained to the data deluge. The high scalability of this storage technology relies on the fact that data are not represented explicitly as relations, and the storage layer is agnostic about the structure of the data [78]. This allows often to model the database as a simple two column table, where a unique key gives access to the structured information, possibly compressed, stored in a unique field. Data are represented, among others, using XML[3], JSON[4] or other proprietary syntax (e.g. Google Protocol Buffer[5]). This configuration is optimal to support horizontal partitioning of the dataset, that can be easily distributed on several machines, providing the required scalability [78]. Database management system becomes then distributed systems, and the access to the records are mediated

[1] http://incubator.apache.org/cassandra/

[2] http://aws.amazon.com/simpledb/

[3] http://www.w3.org/TR/2006/REC-xml11-20060816/

[4] http://tools.ietf.org/html/rfc4627

[5] http://code.google.com/p/protobuf/

novel data processing models such as MapReduce [54], or by using distributed inverted indexes (e.g. Apache Solr [136]), which provide sub-linear scalability with respect to the indexed data.

The ongoing paradigm shift in data representation models is among the reasons to move from a record-oriented conception of the entity matching problem abstracting records to the level of *descriptions*. Descriptions can be conceived as simple *sets of attributes* in form of $\mathcal{A} = \left\{a_1^{[\mathcal{M}]}, ..., a_n^{[\mathcal{M}]}\right\}^{[\mathcal{C}]}$ and $\mathcal{B} = \left\{b_1^{[\mathcal{M}]}, ..., b_s^{[\mathcal{M}]}\right\}^{[\mathcal{C}]}$, where a_i and b_i are attributes of the form $(\alpha_i = v_i)$ with α as possibly empty attribute name and v as attribute value. Furthermore, $\mathcal{M} = \{m_1, ..., m_j\}$ is a set of *metadata* related to the attributed value (e.g. language, encoding, timestamp, etc), and $\mathcal{C} = \{c_1, ...c_k\}$ is a set of contextual parameters referring to the description (e.g. entity type, provenance, etc.). Entity matching consists in attempting to establish whether $\mathcal{A}$ and $\mathcal{B}$ refer to the same real world entity and thus to to assert that $e_1 = e_2$.

The problem of Entity Matching, widely studied in the information system and database community, produced in the year a large set of techniques that often resulted to be effective in their application contexts. Most of the techniques conceived in the information system area relied on a set of assumptions outside of which the matching rarely performed in a satisfying way. Among others, a common assumption is that the matching task is executed on a specific type of entity, after supervised preprocessing task aiming at data standardization, including field names homogenization and data format conversion. Typically, the techniques defined are assumed to be supervised by an expert mastering the domain knowledge underlying the information system, and thus capable of tuning the matching tool and taking adequate matching decision on the cases requiring clerical review. These, and other assumption, often allowed to circumscribe most of the entity matching decision making in the surrounding of the string matching problem for which several sophisticated techniques were defined and successfully applied. However, it is clear that these techniques were conceived to be applied in a controlled environment and under the supervision of human experts that takes responsibility for the quality of the data in the integrated system. Furthermore, any decision about a pairwise entity matching would affect only the aligned databases, without any effect on the world outside the integrated one. An overview of the more relevant approaches is presented in the section 3, particularly in section 3.1.

One of the main promises of the vision corroborating the development of the Semantic Web is the possibility of exploiting the automatic integration of a potentially vast amount of semantically structured information. In particular, recent trends go towards the definition of entity-centric information processing, aiming at producing mesh-up of sparsely distributed information about real world entities, see for example

Sig.ma[6]. This fosters the creation of entity-centric search engines, raising new challenges for the solution of the entity matching problem in the context of the Web. As mentioned in the introduction, the lack of naming convention guiding the adoption of a defined set of globally unique URIs as non-ambiguous means of reference to real world entities, forces the pursue of information integration to pass through the solution of entity matching problem along generally ambiguous descriptions [75]. Unfortunately, many of the traditional assumptions leading to an acceptable solution of the matching problem in the information system context are not valid in the open context of the Web. For example, matching decision can neither rely on semantic and structural homogeneity of data achieved along preprocessing tasks, nor rely on trustworthy human expertises to supervise any ambiguous matching decision as the amount of information available on the Web is in general too broad to be handled and mastered, and thus strongly affected by subjectivity issues. Furthermore, in the context of the Web of Data, the natural conclusion of an objects consolidation task implies the creation of an *owl:sameAs* statement explicitly declaring the *identity* between two resources. Such statement becomes a novel part of the Web of Data, and thus would affect any further processing of the information related to the matched resources. A recent study [73] estimated that only 51% of the currently existing *owl:sameAs* statements part of the Linked Data information space connects descriptions referring to the same real world entity. The estimation was performed by requesting people to evaluate the equivalence of the descriptions of supposedly identical resources. An overview of the most relevant approaches dealing with object consolidation on the Web is presented in section 3.2.

In this context we define *Open Entity Matching* as a reformulation of the traditional entity matching problem in the context of the Web, assuming its scale, mutability, heterogeneity and possible inconsistencies, without making any strong assumption about the quality of the information involved in a matching process. In particular, no assumption is made neither about the way data are structured, given that it can be represented in the very general form described above, nor about the semantic of the attributes composing a description. From now on we refer to these characteristic as semantic and structural heterogeneity of a description. This means that, in principle, a solution for the open world entity matching problem should provide matching decision for any type of entity and considering the Web as the domain of interpretation for all possible matching entities. These settings cause several complications:

1. **semantic heterogeneity**: descriptions are often represented according to different vocabularies and schemas. This problem is typical also of information system ETL tasks. However, database integration is usually supervised by database

[6] `http://sig.ma`

administrators that master the domain considered and are capable of aligning pairwise the schemas involved. This task is intuitively more complicated when managed in an open, wide and heterogeneous context such as the Web. Indeed, it is not uncommon that different people interpret differently the semantic of properties and attributes when they choose how to semantically annotate their data. This natural ontological relativity in the interpretation of the semantics of attributes is one of the causes of heterogeneous usage of attributes.

2. **structural heterogeneity**: attributes are often represented at different levels of granularity. There are descriptions that wrap most of the descriptive information in a generic descriptive paragraph, and others that rely on a wide set of attributes. Specific composite attributes can be represented as a unique field, or specifying each element in different attributes. For example, an address can be represent as a unique field "address" or can be shredded in 'street number', 'street name', 'city', 'zip code', 'state' etc. Similar approach can be applied to other attributes types such as date, name, geo-coordinates, etc. . Another phenomena that can be found in the wild and uncontrolled domain of the Web is the fact that composite attributes are represented as multi-valued instances of the same attribute. For example, an address elements could be represented as several instances of the "address" property.

3. **underspecification**: a description could be underspecified with respect to its interpretation in an open context, possibly omitting implicit contextual information, and thus causing problems of ambiguity (e.g. a description of a restaurant could omit the name of the city among the attributes used for the description, using only on the street name and number) as the data are meant to be available through a web site specific for a city. Another example could be MusicBrainz[7] dataset, where names of artists are mentioned without any specific reference to the fact that they are musicians creating ambiguities when homonym exist in sources providing data about Health Care Providers [8].

4. **over-specification**: a description could be over-specified, presenting an excessive amount of information that is relevant or interpretable only within a specific context, or in general not relevant for identification purposes [111].

Intuitively, the larger is the context considered when taking a matching decision, the higher is the possibility of dealing with underspecified descriptions. For example, considering the limited context of a family, simple descriptions containing only first

[7] `http://musicbrainz.org/`, open music encyclopedia

[8] Factual Health Care Provider dataset `http://www.factual.com/data-apis/places/healthcare`

names would be sufficient to solve entity matching problem. However, if we match descriptions of people in a school, the matching of first names does not suffice anymore, and further information is necessary to take accurate matching decision. The amount of information necessary to take a matching decision in limited context is usually small enough to allow definition of an information model supporting a precise system of unambiguous references (information system database). However, when it comes to the Web, determining the precise amount of information necessary to take an accurate matching decision would require complete knowledge about each real world entity mentioned on the Web, which is practically impossible.

2.1 Examples Of Semantic Heterogeneity

http://xmlns.com/foaf/0.1/givenName: Antônio Carlos Brasileiro de Almeida Jobim

http://xmlns.com/foaf/0.1/givenName: Antônio Carlos

http://dbpedia.org/ontology/abstract : Antnio Carlos Brasileiro de Almeida Jobim (January 25, 1927 &ndash; December 8, 1994), also known as Tom Jobim, was a Brazilian songwriter, composer, arranger, singer, and pianist/guitarist. He was a primary force behind the creation of the bossa nova style, and his songs have been performed by many singers and instrumentalists within Brazil and internationally.

http://dbpedia.org/property/label: MCA Records

http://dbpedia.org/ontology/wikiPageExternalLink: http://www.tomjobim.com.br/

http://purl.org/dc/terms/subject Category:Msica Popular Brasileira pianists

http://dbpedia.org/ontology/occupation: Singer

http://dbpedia.org/property/name: Tom Jobim

http://purl.org/dc/terms/subject: Category:Brazilian singer-songwriters

http://dbpedia.org/property/born: 1927-01-25

http://dbpedia.org/property/associatedActs : João Gilberto

http://dbpedia.org/property/label: Philips Records

http://purl.org/dc/terms/subject: Category:Verve Records artists

http://dbpedia.org/property/origin: Rio de Janeiro, Brazil

http://dbpedia.org/property/name: Jobim, Antonio Carlos

http://dbpedia.org/ontology/genre: Msica Popular Brasileira

http://xmlns.com/foaf/0.1/surname: Jobim

http://dbpedia.org/property/background: solo singer

http://xmlns.com/foaf/0.1/homepage: http://www2.uol.com.br/tomjobim/

http://purl.org/dc/terms/subject: Category:Cardiovascular disease deaths in New York

http://dbpedia.org/ontology/hometown: Rio de Janeiro (state)

http://dbpedia.org/ontology/recordLabel: A&M Records

http://dbpedia.org/property/dateOfDeath: 1994-12-08

http://purl.org/dc/terms/subject: Category:Grammy Award winners

http://dbpedia.org/property/label: Decca Records

Table 2.1: Description retrieved from DBPedia

In order to make more explicit the complexity and multi-faceted nature of the problem, please consider the samples of descriptions presented in tables 2.1 and 2.2. The table 2.1 presents the description of the Brazilian musician Antonio Carlos Jobim (aka

name: Antônio Carlos Jobim

occupation: Musician

occupation: Artist

mbid: 7a8dbe84-f4c0-4457-bfa3-edced1f8cde0

url: `http://www.last.fm/music/Ant%C3%B4nio+Carlos+Jobim`

image: `http://userserve-ak.last.fm/serve/34/2245888.jpg`

image: `http://userserve-ak.last.fm/serve/64/2245888.jpg`

image: `http://userserve-ak.last.fm/serve/126/2245888.jpg`

image: `http://userserve-ak.last.fm/serve/252/2245888.jpg`

image: `http://userserve-ak.last.fm/serve/_/2245888/Antnio+Carlos+Jobim.jpg`

streamable: 1

tag: bossa nova **tag**: jazz **tag**: brazilian **tag**: mpb **tag**: latin

bio_summary: Antônio Carlos Brasileiro de Almeida Jobim (born January 25, 1927 in Rio de Janeiro, Brazil December 8, 1994 in New York City), also called Tom Jobim, was a Brazilian composer, arranger, singer, pianist and perhaps the greatest legend of bossa nova. Jobim's compositions, many performed by Jo&atilde;o Gilberto, gave birth to the genre in the early 1960s. Jobim's roots were planted firmly in the works of Pixinguinha, a legendary musician and composer who, in the 1930s, began the development of modern Brazilian music. He was also influenced by the music of French composer Claude Debussy and by jazz.

album: Finest Hour **album**: The Girl From Ipanema (A Retrospective) **album**: Indito **album**: Finest Hour **album**: Jazz 'Round Midnight **album**: Verve Jazz Masters 13 **album**: Antonio Carlos Jobim em Minas ao Vivo Piano e Voz **album**: Terra Brasilis **album**: The Essential Antonio Carlos Jobim **album**: Wave **album**: Sun Sea And Sand - Favourites **album**: Stone Flower ...

Table 2.2: Description retrieved from LastFM

Tom Jobim) that can be found in DBPedia[9]. The table 2.2 presents a description of the same artist as represented in LastFM[10], a platform for the promotion of music events and broadcast. The description of DBPedia contained 117 attributes, for a matter of space we selected a subset of them. The LastFM description presented is actually complete with respect to what we could find at the moment of writing this work.

Looking at the description, it is possible to conclude that the descriptions refer to the same person. Despite the sets of information only partially overlap, for a human being this task is not particularly complicated. In fact, both descriptions present information about date and place of birth, date and place of death, keywords describing the domain, profession and occupation. However, as it is possible to see, the description collected from DBPedia in table 2.1 presents all the attribute names according to different ontologies, whereas the description from LastFM in table 2.2 structures data according to an XML schema. Therefore, the attributes names presented in the table are the name of the elements and attributes used in the schema. Clearly, the semantic of some attributes can be interpreted in a similar way.

[9] `http://dbpedia.org/resource/Antonio_Carlos_Jobim`

[10] `http://www.last.fm/music/Ant\%C3\%B4nio+Carlos+Jobim`

2.2 Examples of Structural Heterogeneity

Furthermore, the description present structurally heterogeneous descriptions. As a matter of facts, most of the information necessary to take matching decision are described using natural language in the description contained in *bio_summary* attribute for the description contained in LastFM, In particular, the date of birth and date which are intuitively important to take matching decision. Conversely, the description contained in DBPedia presents detailed attributes about these important information.

It is important to notice how simply analyzing the *name* attributes in both descriptions, we can find a certain degree of semantic and structural heterogeneity. The description from DBPedia presents several attributes containing the name of the artist:

`foaf:givenName`: *Antônio Carlos Brasileiro de Almeida Jobim*;
`foaf:givenName`: *Antônio Carlos*;
`dbpedia:name`: *Tom Jobim*;
`foaf:surname`: *Jobim*

Whereas the description from LastFM contains just an attribute about the name:

name: *Antônio Carlos Jobim*

Surprisingly, even if the semantic of the attributes is harmonized, it is not possible to compare the any of the attributes to be equal. Looking more carefully, it is possible to notice that the attributes composing the DBPedia description precisely describe sub parts of the *name* attribute, that are *givenName* and *surname*. Exploiting this syntactic structural knowledge about these attributes, it is possible in principle to define another attribute name for the description in DBPedia:

Antônio Carlos + *Jobim* = *Antônio Carlos Jobim*

In this work, we will focus more on easing issues related to this last time of structural heterogeneity, rather than deal with Natural Language processing aiming at processing textual descriptions to extract possible features embedded in them.

So far, through the description presented in tables 2.1 and 2.2 we highlighted instances of problems related with semantic and structural heterogeneity. However, in the definition of the problem in the previous section we mentioned also the possibility of finding inconsistencies. Consider for example the attributes:

`foaf:givenName`: *Antônio Carlos Brasileiro de Almeida Jobim*;
`foaf:givenName`: *Antônio Carlos*

If we rely on the semantic of the attribute defined in the FOAF ontology[11] the first instance of the attribute *givenName* is incorrect, as it contains the whole complete name of the person.

name: Ferrero, Martin; **name**: Martin Ferrero; **subject**: Category:Miami Vice; **birthPlace**: Brockport, New York; **placeOfBirth**: United States; **subject:Category**: American film actors; **abstract**: Ferrero joined the California Actors Theater in Los Gatos, California. In 1979, he moved to Los Angeles and began to act in Hollywood. He is widely remembered for his role as the ill-fated lawyer Donald Gennaro in Jurassic Park (1993). He was a regular on the 1980s TV series Miami Vice for playing two roles during its run on NBC, ...; **birthDate**: 1947-09-29; **placeOfBirth**: Brockport, New York; **birthYear**: 1947; **birthPlace**: United States; **birthPlace**: U.S.; **label** : Martin Ferrero; **dateOfBirth** : 1947-09-29; **givenName** : Martin; **surname** : Ferrero;

Table 2.3: Examples of Inconsistent (but owl:sameAs) descriptions 1

last_name: Ferrero; **birthdate**: July 13, 1947; **full_name**: Martin Ferrero; **tag:** actor; **domain:**cinema; **short_description**: Martin Ferrero (born July 13 1947)is an American stage and film actor. Ferrero joined the California Actors Theater in Los Gatos, California. In 1979, he moved to Los Angeles and began to act in Hollywood. He is widely remembered for his role as the ill-fated lawyer Donald Gennaro in Jurassic Park (1993), but he has also had other significant roles. He was a regular on the 1980s TV series Miami Vice for playing two roles during its run on NBC, ...; **first_name: Martin**;

Table 2.4: Examples of Inconsistent (but owl:sameAs) descriptions 2

2.3 Examples of Inconsistent Descriptions

Analyzing the three descriptions in table 2.3 retrieved from DBPedia[12], table 2.4 retrieved from Okkam[13] and table 2.5 retrieved from Freebase[14], it is possible to conclude that they are about the same actor, despite there are some inconsistencies. In fact, the description from okkam contains a different date of birth for the actor Martin Ferrero. The descriptive paragraphs and other data about the participation in movies and TV series (e.g. Miami Vice) support positive matching decisions, but not the date of birth. This inconsistency is probably due to errors contained in the sources from where the descriptions were extracted. In fact, the three sources extracted part of the information processing the InfoBox of Wikipedia[15]. It seems reasonable to assume that the description retrieved through Okkam was not refreshed recently, and thus lost possible updates represented in DBPedia and Freebase. Similar error can be found also in PalZoo celebrity database[16]. Thereby, when implementing a knowledge based solution, we have to keep into consideration also these aspects. Formally dealing with inconsistencies is quite complicated, and there exists branches of logic aimed at finding consistent ways to deal with possibly inconsistent knowledge.

[11] `http://xmlns.com/foaf/spec/#term_givenName` The given name of some person. The givenName property is provided (alongside familyName) for use when describing parts of people's names. Although these concepts do not capture the full range of personal naming styles found world-wide, they are commonly used and have some value.

[12] `http://dbpedia.org/page/Martin_Ferrero`

[13] `http://www.okkam.org/eid-dc0d15ca-f887-46b8-a172-0901d0f44859`

[14] `http://freebase.com/view/en/martin_ferrero`

[15] `http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Infoboxes`

[16] `http://www.palzoo.net/Martin-Ferrero`

name: Martin Ferrero; **gender**: Male; **nationality**: United States of America; **description**: Martin Ferrero (born September 29, 1947) is an Am; **Gender**: Male; **series**: Miami Vice; **date_of_birth**: 1947-09-29; **Place_of_birth**: Brockport; **Country_of_nationality**: United States of America; **Episode**: Brother's Keeper; **profession**: Actor; **film**: Stop! Or My Mom Will Shoot!;

Table 2.5: Examples of Inconsistent (but owl:sameAs) descriptions 3

Chapter 3

The State of the Art

This chapter presents a review of the most relevant literature concerned with the problem of entity matching. In particular, the literature review is presented considering two different contexts of application. Entity matching problem had its historical development in the context of databases and information systems. Along the years, efficient and effective solutions were defined to solve the problem in this context as presented in section 3.1. However, considering wide open context of the web, some of the proposed solution do not necessarily apply as many of the assumptions are not valid. Hence, a review of the solutions proposed in this context is presented apart in section 3.2.

3.1 Entity Matching in Information Systems

The most recent survey about the topic found using traditional search engines such as Google Scholar[1] is **Elmagarmid, Iperitos, Verykios**, 2007 [58]. Enterprises rely on information systems whose data, stored in databases, contains duplicates and in general low data quality due to misspelling errors, or different input conventions. The authors present a brief introduction of the data preparation task which includes *parsing*, *data transformation* and *standardization*. Parsing is about isolating individual data elements in the source files. Data transformation refers to conversion that can be applied to data to align data types, e.g. in legacy context. This task includes fields renaming, so that fields from different sources can be compared in a uniform manner. Data standardization refers to the process of standardizing the information presented in certain fields, e.g. addresses, date, time, etc. Prepared data are usually stored in tables, and analyzed to decide which fields should be compared, e.g. it does not make sense to match address with name. The overall data processing is also known as ETL

[1] http://scholar.google.com/

(Extraction, Transformation, Loading)[96].

In [58], the authors of the survey dissect existing methods in two main categories:

- Methods that try to learn how to match using machine learning and probabilistic techniques;
- Methods that rely on domain knowledge or generic distance metric;

In the following sections, we aim at presenting a review of the literature considered relevant about entity matching, providing pros and cons of each of the approaches. The section first provides an analysis of probabilistic methods, followed by an analysis of distance based methods and rule-based solutions. The literature analysis is not aimed to be complete, but to present an analysis of more recent literature considered relevant. For a more complete analysis of existing works, please refer to the surveys [58] and [32].

3.1.1 Probabilistic Methods

The present section presents an analysis of a literature describing probabilistic solutions to the entity matching problem. The first part of the section provides a simple theoretical introduction, mentioning some historical works at the base of many existing solutions. The second part of the section introduces briefly the principles of different machine learning approaches and techniques, and finally an analysis of the most recent papers is presented.

Newcombe et al. [76] and Fellegi and Sunter [60] were the firsts to recognize duplicate detection as a Bayesian inference problem. In particular the record linkage can be conceived as classification problem based on Bayes Decision Rule optimized to minimize the error or, alternatively, the cost of the error.

Bayes Decision Rule for Minimum Error: taken a random records pair, the goal is to determine whether this pair can be classified as *match*, or *nonmatch*. Naively, if the probability of being classified in a class is larger than the probability of being classified in the other is the main driver of the decision. To minimize the error, a Bayesian test for miminum error based on the *likelhood ratio* of the classification is computed. Namely, the *likelihood ratio* of the classification is tested against a likelihood threshold estimated relying on the probability distributions of the datasets. It is proven that this type of classification guarantees is optimal in terms of error minimization [58].

Bayes Decision Rule for Minimum Cost: the minimization of the probability of error is not necessarily the best criteria for classification rules where misclassification of *match* and *nonmatch* can have different consequences. Thereby, it is appropriate to assign a cost to each situation. Essentially, a classification error (misclassification) cost is combined with the probability estimation, and the classification is performed base on the minimum cost for error test. When misclassification costs are symmetrical [2], then the classification for minimum cost behaves exactly as the minimum error classification [58].

Decision with Reject Region Even in ideal scenarios, when the likelihood ratio is close to the threshold the cost of error is high. Thus, Fellegi and Sunter [60] proposed to ad an extra *reject* class apart from *match* and *nonmatch.* This class represents those records that require a 'clerical review' by experts. The more problematic aspect of this approach is to sort the thresholds determining the regions so that no region disappears [60].

Naive Bayes method The main drawbacks of Bayesian classification approach, as any other probabilistic method, is that it relies on knowing *a priori* the *match* and *nonmatch* distributions probability functions necessary to estimate optimally the likelihood threshold, which is rarely the case. A way overcome this problem is to rely on a *Naive Bayes* approach to estimate the *match* and *nonmatch* distribution function, assuming a conditional independence of the random variables[3]. This way, the probability can be estimated using a set of pre-labeled samples (a.k.a. training set). When the conditional independence assumption is not reasonable it is possible to apply *General Expectation Maximization* algorithms [156] that work well under specific conditions for unsupervised classification [157]. In particular, these conditions are related to:

- match frequency above 5%;
- clear distinction between classes of samples;
- low number of typos;
- presence of redundant identifiers among the considered fields;
- also training-set based classifications perform well;

An important factor affecting probabilistic methods is the absence of values in some field (i.e. null values). A possible solution to this type of problems is related to the use of multi-dimensional models such that one of the dimension is used to mark the presence of a field value in both records, see for example [21, 127].

Active learning techniques aims at easing the solution to the problem of creating a training set. An active learner picks subsets of unlabeled data that, when labeled,

[2] the cost of a *nonmatch* classified as *match* minus the cost of *match* classified as *match* is equal to the cost of *match* classified as *nonmatch* minus the cost of a nonmatch classified as a *nonmatch*

[3] Conditional Independence implies that the matching probabilities of different fields in the same record are independents.

will give highest information for classification [48]. Some authors proposed to seek for samples lying in the 'reject region' for the creation of the training set [131]. Namely, the individuation of a 'reject region' allows the labeling of the ambiguous cases reducing the size of the training set. Alternative solutions relied on a committee of classifiers to spot the samples classified differently by different classifiers, and thus requires for expert review [143]. Essentially, with few iteration over a dataset, the learner is capable to individualize the more interesting samples and learn about peculiarity of a dataset. However, as mentioned in [67], these techniques may also affect negatively the quality of the match.

Furthermore, in [67], Goiser and Christen propose a critical discussion of supervised probabilistic methods, pointing out the following issues:

- match rarity in large datasets can make the definition of adequate training set cumbersome;
- matching comparison techniques allow to tune parameters such as thresholds. This affects the quality of match and requires a high degree of knowledge about both the comparison technique and the data. Such knowledge is usually time expensive to acquire and seldom reusable;
- tuning of parameters can be learned, but the problem of training set generation remains;
- the adoption of active learning techniques to ease the generation of training set can cause bias possibly affecting quality of classification;

Thereby, Goiser and Christen in [67] propose to rely on unsupervised machine learning techniques combined with effective blocking techniques. The author try to remove the elements criticized proposing to apply adaptive 'on the fly' blocking techniques, and performed experiments analyzing what active learning technique works better. The result of the experiments are positive, even thou the dataset was partially synthetic and referring mostly only to people. The authors proposes an experiment that is parameter free, to see whether it is possible obtain results comparable with methods that use parameter. The authors rely on FEBRL, a bio-medial record linkage system for comparison, and on Weka for classification. The experiment show particularly good results on synthetic data using unsupervised machine learning techniques.

Doan, Lee and Han in [57] propose a Profile-Base Object Matching (PROM) that explores the fact that disjoint attributes are often correlated and can be used to improve matching accuracy. For example, if two tuples present the exactly the same name and surname, but one has the salary attribute set at 100000$ and the other has the age attribute at 6, there is evidence that the two tuples do not refer to the same real world entity. This sanity check is performed using modules that apply different profilers to the matching pairs. The author propose the definition of hard profiler, which represent hard constraints. For example, the review year of a movie cannot precede the release year of the movie. These profiles can be manually defined or automatically learned assuming data are complete. The author further propose the usage of soft-profiler that are likely to be respected, but not necessarily. Soft-profiler are essentially classifiers, given a set of matching and non-matching pairs it is possible to learn such profilers. The authors both explored manually defined and automatically learned constraints. The idea proposed in the paper is excellent, but the evaluation based on a manually created dataset was not really convincing. Furthermore, the approach used for the definition of constraints does not appear to be general, even thou the principles of exploring domain knowledge to improve matching accuracy is correct. The manual definition of profilers is suitable for a closed domain, but can be hardly applied in a open inconsistent context.

Ravikumar and Cohen in [127] introduce a novel approach to unsupervised learning techniques for record linkage. In particular, the authors propose a way to reduce complexity of learning generative models by applying semantic and monotonicity constraints. Semantic constraints essentially move to the 'matching fields' layer the decision whether the whole record matches. Matching fields layer is a layer of latent variables that depends from the feature vector of each pair. However, this approach allows the definition of dependencies between latent matching variables, rather than on feature vector layer. This reduces the number of parameters to be automatically learned. However, the authors did not test the option where dependencies are manually established, and actually flattened the semantic interpretation of matching to the probability that all fields-pair would match to decide about a matching record. This does not assume that the single field-pair matches are error free, indeed feature vector discretization and distance measure are still quite error prone. However, the concept that match should be defined on base of matching attributes is interesting.

Shen, Li and Doan in [133] described a probabilistic solution to entity matching that exploits such constraints to improve matching accuracy. At the heart of the solution there is a generative model that takes into account the constraints and provides well-defined interpretations of them. Real world applications often have many semantic integrity constraints that can be exploited. Such constraints can either be learned or specified by a domain user. The authors propose to use relaxation labeling, previously used in many classification problems. This technique has the advantage of scaling easily on large datasets and that it can accommodate a wide range of domain constraints. The first layer clusters mentions into groups and exploits constraints at group level. Once this is done, the second layer exploits additional constraints at the level of individual matching mention pair. Once automatic matching is performed, a user might want to examine the result, and provide some feedback that is modeled as temporary domain constraints and then re-run the matching. The experiments are focused on one type of entity, but the solution approach can be generalized for many types of entities simultaneously. The proposed solution provides an optimal trade off among human effort, application of domain knowledge, and exploitation of generative probabilistic models. However, the solution seem to be suitable in limited domains, and hardly applicable in an open context.

Zhao and Ram in [160] propose an analysis about the combination of multiple classifiers for the realization of entity identification tasks. The author studied the cascade and stacking combination of classification techniques, and evaluated also the bootstrapping methods used to create the datasets. In particular Bagging and Boosting approaches were evaluated. Bagging methods leads to better performances for unstable classification techniques with rare degradation of performances. Boosting works better than bagging, but it is more sensitive to noisy datasets and thus works better on linear classifiers.

Singla and Domingos in [134] propose an approach to entity resolution based on Markov Logic, that is a sort of approximation of first order logic with probabilistic graphical model. This approach enables the creation of potentially inconsistent knowledge bases supporting anyway automatic inference over part of this knowledge base. The author propose to model the entity resolution problem with a Markov Logic Network with formulas and weights. The most likely truth assignment is computed using MaxWalkSAT and conditional probabilities of query atoms given the evidence are calculated using Gibbs sampling. Equivalent and reverse equivalent relations are defined, and on base of this, it is possible to define some sort of 'logical regression' estimating whether two records are the same on base of words or n-grams (i.e. if two field have the same n-gram/word then there is an evidence that they might be about the same entity). the idea is interesting because it combines expressiveness of first order logic, and enables at the same time machine learning, adaptive, approaches in the definition of the weights used to decide matching classification.

Rastogi, Dalvi and Garofalakis in [125] propose a framework for scale entity matching solution on large datasets. The authors propose to split the dataset into neighborhoods and a message passing protocol to build global solution. The authors relied on Markov Logic entity matcher described in [134], providing a matching rule result of domain knowledge. In particular, the authors evaluated the scalability of their approach on bibliographical dataset, relying on the soft constraint related to co-authorship. The proposed framework was proven to be very efficient when executed on a cluster of 30 servers. The main contribution of the authors is related to scalability of the process, rather than the mere solution of the problem itself.

3.1.2 Distance-based Methods

A way to avoid the problems of machine learning/probabilistic techniques is to define distance metrics for descriptions that do not require any training. Distance-based approaches essentially consider each record from a syntactical perspective and try to compute similarity of records according to one, or a combination of distance metrics and weights [55]. The definition of a similarity threshold is then necessary to establish matching decisions. However, understanding what is the best metric and what are the right weights and thresholds are affected by the same issues of probabilistic and machine learning methods. Namely, they require high degree of domain knowledge to select the proper metric, and tune correctly the thresholds. Another distance-based approach proposed in [71] is based on ranked list merging. That is, every record is compared with the others only considering one field and then the best matching is ranked on top. Comparing all the fields in this way, we get a number of ranked lists to be merged containing the minimum aggregate rank distance. One problem affecting any distance-based technique is that it must rely on properly defined matching threshold. As previously mentioned, relying on training data would nullify the advantages of distance-based metrics, thus alternative approaches were pursued. In [40] it is proposed on relying on a clustering algorithm based on the assumption that records about similar entities usually have very small distance, and that only a small number of records fits within this small distance. So, similarity threshold can be computed for each record improving results of methods relying on predefined threshold.

Churces and Christen in [43] propose a protocol for minimal-knowledge n-gram comparison enabling record linkage between databases without disclosing explicitly any information. The work relies on hashing algorithms used to encrypt values of attributes which are sent to a third party responsible to compute matching distance. The protocol described is quite complex, and shows how it is possible to perform n-gram distance-based record linkage without disclosing information. However, the matching process is based on the assumptions about prior knowledge of what attributes have to be matched.

Bhattacharya and Getoor in [15] propose an approach for record linkage that takes into account the similarity of linked object, without assuming that linked object have been already de-duplicated. In fact, the authors consider links among object of the same type, so when two record are discovered to refer to the same individual it is possible to perform further inference iteratively. Iterative process produce more accurate results decreasing false positive rate, and allows more conservative string matching. The price for this improvements is an increased computational cost as the matching algorithm as to face matching of sets of sets, and furthermore the recursive definition of duplicates adopted leads to an iterative algorithm. The paper focuses on author resolution problem in bibliographic repository context, where references to the same person very frequently, and furthermore there is the problem of homonym. The authors propose to make use of additional context information (e.g. co-authoring information) to understand whether two papers are written by the same author, but this comparison presupposed that the matching authors are already known. The problem of author resolution is likely to be an iterative process as the identification of common authors will allow the identification of further potential co-references. The proposed algorithm starts by clustering references that whose distance is negligible. Then, candidate similar cluster pairs are chosen and iteratively the algorithm evaluates the distance for the candidates, selects the closest pair according to distance measure, merges the clusters and updates the attribute means. The paper presents an approach to record linkage aiming at considering contextual information to improve matching quality through an iterative algorithm that, on base of new matching discovery, upgrades also distance of others candidate matching candidates and attempts to compute further matching. Essentially, the matching is performed also on base of other information associated according to some target relation (e.g. co-author), but those information must be matched as well. So, when new matching is discovered among 'co-authors' for example, then also the distance between the other matching candidate must be updated. The idea is good, but there is no explanation about how to isolate contextual information and how to weight their relevance with respect to the matching.

Bhattacharya and Getoor in [16] propose a different approach to entity resolution, considering relational information among records in the databases, defining thus a collective entity resolution method aiming at discovering co-occurrences jointly rather than in a pairwise fashion. The authors first distinguish between the 'identification' and 'disambiguation' problem, then rely on the concept of 'entity cluster' to implement a *relational clustering algorithm* for collective entity relational entity resolution. Subsequently, the authors evaluated the new proposal for entity resolution performing experiments on real world bibliographic datasets such as Citeseer, arXiv and Biobase. The idea that iterative process and relations can improve precision is very interesting.

Chaudhuri, Sarma, Ganti and Kaushik in [39] propose to exploit aggregate context dependent constraints to accept or reject de-duplication steps produced by record textual similarity to reduce partitioning search space. Constraints are partitioning functions, where partitions are said to be satisfied when all tuples satisfies constraints. The authors sustain that the problem is semantically and computationally hard, so they propose a way to reduce the problem to a maximum satisfaction variant of the problem. The constrains considered are: constraints on individual tuples, de-duplication parameter constraints, pairwise positive and negative examples, and groupwise constraints. These constraints are not applied as hard constraints, but in a little relaxed fashion so that the constraints satisfaction can be expressed as a maximization problem based on a benefit function. The integration of constraints with textual similarity passes through the definition of a similarity graph based on syntactic similarity such that edges are drawn between tuples if similarity is above a certain threshold. The constraints verification is a NP-Hard problem in the size of the set of tuples considered, thus it is necessary to restrict search space avoiding the complexity problem. The author propose to group all the tuples in a unique partition, and the iteratively split the groups based on an increasing similarity threshold. Thus, when an edge has similarity below a certain threshold it is removed, and this is repeated until the group of tuples is disconnected from others defining a space of valid groups. The 'frontier' groups (i.e. the subgraph of the split tree on base of threshold that are just above the leafs) are considered the valid groups. Thus, the de-duplication problem becomes: among all the frontiers, find the one that maximizes the benefit functions. The proposed solution is surely valid but it can be applied in a closed and controlled domain. In fact, most of the computational complexity the authors have to deal with are related to the de-duplication approach considering the whole dataset to compute iterative partitioning. Further, it is not clear how thresholds of constraints should be defined.

Bhattacharya and Getoor in [17] introduce a new perspective to the problem of entity resolution at query time. The author propose a query expansion model to cluster duplicates in a database based on collective entity resolution. The main idea behind collective entity resolution is that solving related entities can help in solving the primary one. In particular the author explore the co-author relation aiming at disambiguating authors in paper repositories. This relation-based matching allows to enrich and mitigate errors due to 'attribute based' syntactic matching only. Nevertheless, the authors do not make any analysis over the type of relations and the evaluation of ambiguity if not in terms of relative dataset. Furthermore, no analysis is performed on ambiguity of queries, and a satisfactory answer is considered an answer that simply returns answer set correctly partitioned according to correct entities. Thus, the approach handles query-time de-duplication, but does not deal with identification/matching issues.

Benjelloun, Garcia-Molina, Menestrina, Su, Whang and Widom in [10] propose a generic approach for entity resolution, where matching and merging methods are treated as 'black boxes' and where rather important properties of the outcome of these methods are considered. The desirable properties of pairwise matching methods are *idempotence*, *commutativity*, *associativity* and *representativity.* The authors formally define the entity resolution operation, and further specify optimal properties of the methods used to merge records presenting information about the same entity. Finally, the authors present entity resolution algorithms satisfying on different levels the properties previously defined, and evaluated them performing comparison between Yahoo!Shopping and Yahoo!Travel datasets. The F-Swoosh algorithm relies on features to define match between records, the concept recalls the "equational theory" defined in [79]. Despite the authors do not explore a theory for the definition of features, this can be considered a useful framework to ground feature based matching and extending the concept of feature to the more rigid and precise concept of fingerprint.

Whang, Benjelloun and Garcia-Molina in [153] propose a modified version of the entity resolution approach presented in [10]. The authors introduce the concept of 'negative rules' to remove potential inconsistencies in the databases after entity resolution is applied. Such inconsistencies might be introduced along pair-wise entity centric record matching and merging processes due to the defects of the adopted algorithms. The negative rules essentially analyze records, or group of records, and state whether these are consistent or not. This type of rules cannot be 'injected' into matching and merging methods, and thus must themselves be treaded as black boxes with specific properties that would make them suitable for improving accuracy of entity resolution process. The author defined two approaches for the application of the negative rules defined, implemented and evaluated them on Yahoo!travel dataset. The authors propose an early approach to solve inconsistencies in the data supported by domain expert solver. Given the supervised natured of the process, the human load of work for the application of negative rules is a factor of evaluation. The principle of using negative rules to improve precision of matching is not new, but the formalization proposed by the authors is convincing.

3.1.3 Rule-based Method

An identity rule for a set of real world entities E can be logically defined in this way: $\forall e_1 e_2 \in E, P(e_1.A_1, ..., e_1.A_m,\ e_2.B_1 ... e_2.B_n) \rightarrow\ (e_1 \equiv e_2)$ where P is a conjunction of predicates on the attributes $A_1, ..., A_m$ and $B_1, ..., B_m$ respectively. Furthermore, for each attribute of the conjunction predicate $e_1.A_i \approx e_2.B_i$. If it exists, a rule capturing an identifying attribute is of the form: $\forall e_1 e_2 \in E, (e_1.A_k = e_2.A_k) \rightarrow\ (e_1 = e_2)$. On the opposite, distinctive rules are of the form: $\forall e_1 e_2 \in$

$E, P(e_1.A_1, ..., e_1.A_m, e_2.B_1, ..., e_2.B_n) \rightarrow$ $(e_1 \not\equiv e_2)$. Ideally, the matching should be monotonic, in the sense that adding of information must not change the matching classification. The more relevant works found on rule based methods for entity matching in information systems are [103] and [79] here summarized.

According to Lim, Srivastava, Prabhakar and Richardson in [103], entity matching is to determine the correspondence between object instances from more than one database. The authors propose the use of extended key which is the union of keys from the relations to be matched, and its corresponding identity rule, to determine the equivalence between tuples from relations which may not share any common key. In a single database context usually one object instance can model a real world entity. This is not true for different databases, causing breakdown of information model [93]. The independent development of the databases results in databases capturing different parts of the real world. This makes difficult, if not impossible, to provide integrated access to the databases. Logical heterogeneity can happen at schema level or instance level. The latter can be performed only when schemas are semantically compatible but instances corresponding to the real world are not identified and merged. *Entity identification is the problem of identifying object instances from different databases which correspond to the same entity.* In this paper the authors investigate the use of extra semantic information to partially automate entity identification process. Two tuples from different relations are said to match if they model the same real world entity. Synonym problem arises due to the fact that attributes in both relation are not semantically equivalent (e.g. if the same employee is give different employee number), while homonym may arise also because the key of the relation is not a key in the integrated world (e.g. the name of an entity). The authors model the entity matching process as a three-valued function which takes a pair of tuples and returns TRUE if the pair refer to the same real world entity, FALSE if does not refer, UKNOWN otherwise. Record pairs can be represent in *matching table* and *negative matching table.* The *matching* and *negative matching tables* must respect uniqueness constraint (i.e. no tuple in either relation can be matched to more than one tuple in other relation), and consistency constraint (i.e. no tuple pair can appear in both the matching and negative matching table). The matching function must respect soundness constraint: each record pair declared to be matching (non-matching) indeed models the same (distinct) real world entity (entities); and completeness constraint: the entity identification process returns a value of matching or not matching for all pairs of tuples. To differentiate the analyzed tuples, an attribute denoting the source is appended to the tuple (e.g. 'DB1'). This allows the definition of rules specific for particular databases. To achieve soundness, all information used for entity identification must be correct with respect to the integrated world. Rules defined by administrator are used for define identity and distinctness. An identity rule for a set of real world entities is of the form: $\forall e_1 e_2 \in E, \bigwedge(e_1 \cdot A_1 ... e_1 \cot A_m l, e_2 \cdot B_1 ... e_2 B_n) \rightarrow (e_1 \equiv e_2)$. Furthermore, for each attribute of the conjunction predicate $e_1 \cdot A_i = e_2 \cdot A_i$. If it exists, a rule capturing the identifying attribute can be of the type: $\forall e_1 e_2 \in E, (e_1 \cdot A_k = e_2 \cdot A_k) \rightarrow (e_1 = e_2)$. A distinctness rule for a set of real world entities is of the form: $\forall e_1 e_2 \in E, \bigwedge(e_1.A_1, ..., e_1.A_m l, e_2.B_1, ..., e_2.B_n) \rightarrow (e_1 \not\equiv e_2)$. To guarantee completeness, it is required that enough information is available. This might mean that complete knowledge about entities is required, and such amount of information is often impossible to achieve. Furthermore, in order to guarantee soundness of the entity identification process, the technique must be monotonic. A monotonic entity identification process is such if for every pair of tuple matching or not matching, this classification remains so when additional information is supplied. The authors propose a sound entity identification technique based on the concept of Extended Key Equivalence and Instance level Functional Dependencies. *Extended Key*: is a minimal set of attributes needed to uniquely identify an instance in the integrated real world. The identity rule is then defined on base of Extended Key constraint in order to guarantee that the tuples satisfying the matching condition are unique in their relations. In order to identify attributes that could correctly extend the set of keys of a tuple the authors propose the introduction of Instance Level Functional Dependency (ILFD). ILDF is a semantic constraint on the real world entities that imply some conclusion. If an entity as certain value in certain attributes, this implies it has also some other property. For example, with some background knowledge, it should be possible to state that if a restaurant is specialized in 'spaghetti' then it offers 'italian' cuisine.

Hernandez and Stolfo in [79] deal with instance identification problem known as “merge/purge”. This problem is closely related to a multi-way join over a plurality of large database relations. These strategies assume a total ordering over the domain of the join attributes (an index is thus easily computable) or a near perfect hash function that provides the means for inspecting small partition of tuples when computing the join. Unfortunately, open context cannot rely on all these features because data supplied by various sources typically include identifiers or string data that are either different among different datasets or simply erroneous due to a variety of reasons (including typographical errors or fraudulent activities). To determine whether two records from two databases provide information about the same entity, rule based knowledge base is used to implement equational theory. The author presents a system that performs merge/purge process, including declarative rule language for specifying equational theory making it easier to experiment and modify the criteria for equivalence. Alternative algorithms that were implement for the fundamental merge process are comparatively evaluated, and demonstrate that no single pass of the data using one particular scheme as a key performs as well as computing the transitive closure over several independent runs each using a different key for ordering data. The moral is that several cheap passes over the data produces more accurate results than one expensive pass over the data. The inference that two data items represent the same domain entity may depend on considerable amount of statistical, logical, and empirical knowledge. Wrong entity identification can be worst that missing some matching data. The accuracy of the result is very relevant. The author proposes the sorted-neighborhood algorithm to discover matching record. One obvious way to bring together clusters[4] of records close together is to sort the records over important discriminating key attributes. The effectiveness of this sorted neighborhood method relies in the quality of chosen keys used to sort. The algorithm consists of three steps: create keys, sort data, merge data performing matching evaluation within a shifting window. Notice that the extraction of keys might be an expensive task. To improve efficiency the author propose to perform clustering of data first, and applying sorting neighborhood algorithm only on clusters. However, real world data might not be uniformly distributed, thus clusters might have different size, and either be very small or very big. To establish matching, it is necessary to define equational theory that dictates the logic of domain equivalence, not simply value or string equivalence. For this reason, the author defined a declarative rule language requiring inference over large datasets. The effectiveness of the sorted neighborhood depends on the key selected to sort the records. A key is defined to be sequence of subsets of attributes or substrings within the attributes, chosen from a record. In general, no single key will be sufficient to catch all the matching records. Attributes that appear first in the key have a higher priority than those that appear after and if errors happen to be in an important part of the key then there is little chance to perform correct matching. To compensate this fragility of the single key based matching, the author propose a multi-pass strategy, running several time sorted neighborhood algorithm using a different key all the time. This approach reduces the effect of errors in data, as unlikely all field will contain error in key attributes. The authors perform experiments to achieve both improvements in computational time and precision. The conclusion is that clustering does not produce improvements accuracy and modest improvements in computation time. The multi-pass approach applying several keys and then computing transitive closure produces the best accuracy and decreases the number of false positives. The approach to identification proposed by the authors seems more suitable for a robust entity matching, as it would rely on some

[4] complete comparison is not feasible in large datasets

kind of rigid inference embedded in a background knowledge base guaranteeing, when equivalence rules are satisfied, a precise match. The main drawback of this approach is that precise rules are hard to define, and usually requires a considerable amount of human effort.

Ganesh, Srivastava and Richardson [61] in propose an attribute value distance based approach for learning identification rules in database context. Basically, a training set is provided allowing a learning to associate distance between values to matching results. The learning process produces a decision tree, that at leaves present the identification rules. Evaluation on 1000 of records showed precise accuracy.

Wang, Li, Yu, and Feng in [149] propose a method for learning what similarity metrics works better for discovering duplicates in a dataset. The authors proposed an efficient optimization method capable of learning what similarity metrics works better, reducing the search space for the optimal threshold to a problem of redundancy removal. The authors evaluated the proposed solution on the bibliographical and restaurant dataset, with results comparable with optimal classifier such as SVM.

Chen, Jin, Zhang and Zhou in [41] propose a method for learning matching rules and relative thresholds relying on machine learning techniques to cluster groups of attributes and greedy maximization algorithms to optimize matching thresholds. The authors experimentally evaluated the approach on bibliographic references and restaurant dataset. The proposed produced satisfying results on the analyzed datasets.

3.2 Entity Matching in the (Semantic) Web

An increasing amount of semantically structured documents are produced in the context of the Linked Data development. Along this process, the tendency is to deliberately allow the proliferation of identifiers for entities, relying on the assumption that with time conventions will emerge. The result is that entity matching is not anymore a local pairwise problem, but a large scale distributed and uncertain data management problem [50]. For example the proliferation and mixture of standards for the definition of personal information (FOAF[5], hCard, DBLP, etc), created a mixture of diverse machine readable profiles. The integration of such information would be very useful, but unfortunately this process is not easy due to the fact that profiles rely on different identifiers and other fuzziness, causing a break down of the information model similar to the one analyzed by Kent in the database context [93]. The entity matching problem in this context is known as object consolidation, and several techniques have been recently developed.

[5] Friend Of A Friend, `http://www.foaf-project.org`

Tajeda, Knoblock and Minton in [143] propose an active learning method based on a committee of classifiers to select problematic samples to be labeled by a domain expert. The training set so formed is then used to learn object matching rules to be applied on semi-structured data available on the Web. The idea of relying on a committee of classifiers to learn effective rules is brilliant, but the learned classification rules where conceived to match pairs of dataset in a fuzzy but known and controlled context (i.e. restaurants in Los Angeles), and thus the robustness of the method applied as general solution has to be further studied.

Michalowski, Thakkar and Knoblock in [108] present a potentially very useful approach to object consolidation process. Indeed the author propose to rely on secondary sources for disambiguating uncertain matching provided by a classifier. The authors do not explore deeply the implications of such work presenting solutions that sounds more ad-hoc than general. Surely, the principle of relying on domain knowledge and external resources to decide about object consolidation opens many opportunities compared with traditional closed object consolidation systems. In fact, machine learning classifier for object consolidation work as good as the training set given. The more critical cases need to be treated in a more compelling manner, and opening the matching process to consider secondary sources could help in providing better match/non-match decisions.

Minton, Nanjo, Knoblock, Michalowski and Michelson in [112] propose a record linkage approach based on a combination of 'expert system' and machine learning. Essentially, the matching performed by defining a set of transformation rules that allow to establish matching on a string level. These transformation rules are ranked in terms of “relevance” with respect to the matching problem. Also a set of global transformation are defined, labeling the fields in terms of frequency or semantic annotation (this aspect is only marginally treated). Furthermore, given a training set, the average probability that a certain transformation is applied to a record field is computed. The combination of the transformation applied to each field builds a transformation graph, that defines the transformation of a record to be the same as another one. This transformation graph is then used to compute a similarity measure (distance) to classify the match/non-match of a record couple. The training set will highlight the combination of transformation (transformation graph) that are more likely to lead a match or a non-match, and this probability is computed, normalized and used as similarity score. The idea goes beyond traditional string matching, but is still affected by semantic issues due to the fact that semantic of fields is neglected. Namely John Smith and Smith John can be the same person or not, depending on which one is the name and the surname of the person. From a string matching perspective these can be perfectly the same, but it does not mean that they are. However, the solution proposed tackles effectively the problem of matching descriptions presenting heterogeneous representations of values for the same field.

Bekkerman and McCallum in [9] propose a framework for disambiguating people appearance on the Web considering specific social networks. The authors describe a method overworking the specific link structure of Web pages to define a sophisticate agglomerative/conglomerative clustering algorithm to disambiguate web pages presenting information about an entity from web pages presenting references to homonyms. The work presented by the authors can be described as a generic method for performing unsupervised entity matching on the Web given a social network context (e.g. a mailing list). The solution proposed by the authors is promising, but cannot be applied as a general solutions as it would need a global social network as source for disambiguation.

Hogan et al in [82] propose a scalable method for object consolidation based on a sort of ad-hoc reasoner implemented relying on efficient data structures as e.g. inverted index. The proposed algorithm processes all sets of quadruples (RDF triple (s, p, o) + context, i.e. source) discovering equivalence based on the "inverse functional" meta-property of predicates p used in the RDF triples. If the object o of such predicates p are the same, then the subjects s are stated to be equivalent. Once determined the equivalence between 'subjects' s, instances are also consolidated relying on the consolidated identifiers. The paper presents a linear solution to a problem, but does not explore the possibility that errors are actually possible and thus that the value of inverse functional properties might not be really inverse functional, or that diverse values could be used for the same property referring to the same object. The solution appears to be scalable, but not optimal.

Wu et al. in [158] propose a method for entity matching based on analysis of publicly available documents. In particular the authors focus on the identification of persons in the obituaries relying on a set of attributes considered sufficient to identify a person. This is a specific case of entity resolution, where entity matching in databases is not performed for consolidation, but to remove the records of decedent from diverse databases (e.g. phone book). The proposed method consists in creating lists of candidate identities based on partial set of identification attributes, and then resolve identity searching for equivalences among these lists considering the complete set of identification attributes. High quality automatic data integration is a topic that is receiving increased attention due to work in Semantic Web or more general-purpose web information system. In this paper is presented a novel approach for entity resolution, that combines probabilistic and ontological methods for computing the matching probability of two descriptions.

Ioannou, Niederee and Nejdl in [86] propose a probabilistic method for entity linkage in the heterogeneous information space. In particular, the authors propose to solve the entity matching problem relying on Bayesian network to model properties of entities and their interdependences. According to the authors, this allows to better accommodate information changes in the information space. A network model is built incrementally linking entities through relations that can be direct (namely the entities' descriptions contain the same object), or deduced (namely result of a deductive process due to the combination of other attributes. The model is then used to compute the probability that two entities are actually matching or not. Matching decision is then taken comparing the probability with a threshold. The proposed method is robust and funded on well established mathematical tools. However, in a global space, the problem of establishing the probability threshold for establishing matching decision persists.

Castano et al. in [37] proposes $HMatch(I)$, an ontology instance matching approach based on recursive identification of relevant individuals composing a novel data instance in ontology evolution context. Essentially, every ontology instance in an ABox is represented as a tree of properties and values. This tree is then used for comparison exploring similarity for nodes where individuals (representing other instances) have properties. Properties are weighted differently to discern the ones that are relevant for identifying entities. Considering descriptions as wholes (i.e. trees) can help in providing precise matching of instances as all the properties available in the knowledge base are considered. However, the definition of weights for identifying attributes is considered only superficially.

Mauroux et al. in [50] present IdMesh, a system based on a distributed probabilistic graph defining an overlay above declarative links between entities (web objects) and their referents. The main goal is to analyze such graph to understand how reliably an identifier (URI) can be used to refer to a real work entity. Indeed, in the Semantic Web there is a proliferation of identifiers and connection between them that would make hard to understand how to use them. Voltz et al. in [147] propose Silk Linking Framework, a platform for the definition and maintenance of links between datasets in the Web of Data. Silk offers a discovery engine, a link evaluation tool, and a protocol for link maintenance. The link specification language allows, together with the set of built-in similarity measures, the definition of equational theory enabling the discovery engine to define links between datasets. The main drawback is that the framework is totally supervised and assumes knowledge about schemas of target datasets for the definition of linkage rules. In SILK, linkage rules are conceived as single, or combinations of, attribute(s) matching scores, where the matching threshold is manually defined by the user.

Glaser et al. in [66] describes a system to manage co-references in a centralized repository, which presents several advantages compared to the usage of *owl:sameAs* statements as part of the Linked Data. The approach has several practical positive effects and seems to be scalable in principle. The author did not address the problem of discovering co-references, but deals just with their management.

Hogan et al. in [81] propose a method for consolidating entities in the context of Linked Data. The method consists in crawling the linked data to gather a large number of information in form of RDF triples. Such triples are then processed to infer statistical properties over properties and their values. In particular, it is attempted to infer whether certain properties are inverse-functional or functional on base of the their 'usage' in the dataset. The authors propose several refinements to a very naive approach, producing promising primary results. Basically, cardinality of couples subject-property and property-object is used to estimate similarity of properties, and then these are used to produce an aggregate consolidation confidence. Furthermore, the authors outline a potentially scalable system. The attempt of estimating meta-properties of values relying on statistical methods compensates the unreliable usage of ontologies in the Linked Data context. However, for how scalable is the system, the solution conceived is limited to the processable samples.

Kopke and Rahm in [99] propose a framework for Self Tuning solution for Entity Matching (STEM). The STEM is framework for Entity Matching providing libraries of classifiers, blocking methods and similarity metrics. The framework then takes in input datasets, and provides two methods for selecting samples to be labeled by expert. The combination of different classifiers, similarity metrics and blocking system defines a matching strategy (aka matching process) which includes the possibility of using multiple classifiers and take majority vote decisions, etc. The authors then evaluated the two methods for sample selection, showing how equal proportion of matching/non matching samples in a dataset provide better classification.

Zaho and Rahm in [161] propose an entity matching method based on the constrained cascade application of decision trees classifier to reduce the bias of the simple usage of the classifier. Essentially, cascade application of decision trees (or other classification methods) forces every branching decision to be taken considering a multivariate test that considers the features learned by the cascading classifiers. The application of cascading classifier has to be constrained to some criteria. Accuracy constraints seem to be prone to over-fitting issues, whereas constraints maximizing the depth of the learned three seem to provide the best solution to capture the complexity of data. This approach reduces the bias of a single classifier, outperforming it capability of classification. The authors evaluated the proposed solutions manually choosing similarity metrics for each of the attribute types and in an environment with controlled vocabulary.

Stoermer, Rassadko and Vaidya in [139] propose a novel approach for entity resolution, that combines probabilistic and ontological methods. This work explores the approach presented in a previous paper [138]. The author defined an ontology presenting six top level entity types, each of which is tied to two types of features: (1) generic features (name, type) that are type independent; (2) type-discriminative features that helps to infer an entity type from its description [5]. These features were then associated with an estimation of their relevance along an identification process, and employed to perform general purpose entity matching. The relevance of the type-discriminative features was estimated by performing experiments involving human users [4].

Isele and Bizer in [88] propose GenLink, a solution to learn expressive entity matching rules relying on regression function based on genetic programming. The proposed solution aggregates the selection of attributes to be compared, transformation functions, similarity metrics and relative thresholds to produce expressive linkage rules. The authors performed experiments evaluating results over OAEI datasets. The solution proposed by the authors performed better than the one participating at OAEI and also compared to existing genetic programing solutions applied to record linkage [52]. Regression functions based on genetic programming are exceptionally effective in learning how to classify data in a dataset. However, they are non-linear solution needing to work in a controlled environment. However, the authors did not evaluate whether rules learned in a dataset can be applied to others. This makes the solution optimal for pairwise dataset matching, but its outcome can hardly be used out-of-the-box for matching purposes.

Niu, Rong, Want and Yu in [118] provide a semi-supervised method for learning matching rules for entity matching in pairs of dataset. The method proposed combines statistical methods to mine properties equivalence based on the values found in the datasets, and what the authors called Inverse Functional Properties Suite (IFPS). IFPS corresponds to conjunction of properties providing altogether support for positive matching decision. The approach proposed gets in input a set of labeled samples used to extract IFPSs and properties equivalence using statistical methods. Then, an iterative Expectation Maximization algorithm is run on the complete mining of rules and matching iteratively. The process is tuned to converge, and was tested aligning DbPedia with Geonames, Geospecies and LinkedMDB. The method proved to be precise, but suffered a recall problem. This is probably due to the semantic and structural heterogeneity of the sources that limits the property equivalence discovery and consequently the mining of the rules.

Wei, Jianfeng, and Yuzhong in [84] propose a combination of statistical methods considering similarity with ontological properties to resolve object co-reference in the Semantic Web. The self training approach consist in process that starts from a kernel of 'owl:sameAs' properties, functional and inverse functional pairs of property/value, and max cardinality constraints. These are used to mine further equivalence that are then used to further train the system iteratively. Functionality and inverse functionality of pairs of property/value are estimated using statistical methods similarly to what proposed in [81]. The method was tested on OAEI 2010 person datasets and on large datasets. The property matching values in the paper seemed to be ad-hoc and entity type based. For example, the authors assumed an association between 'rdf:label' and 'foaf:name'. This association is valid for some cases, but 'rdf:label' is a generic attributes, and its adoption as a placeholder for name is valid just considering DBPedia. Furthermore, the kernel initialization relies on owl:sameAs which are known to be often wrong [74] and, if manually checked correspond to the creation of a training set. The method is promising, but the feeling is that the evaluation is not representative of the real applicability of the method as a general solution.

Ngomo in [117] propose a time effective method for linkage discovery integrated into the LIMES framework. The link discovery is based on the computation of a distance metric that is the result of the combination of the similarities between sets of pairs of attributes. Such distance has to be compared with a threshold to take matching decision. The author propose a formal grammar to describe matching process so that atomic operation could dissected and allow the application of time-efficient algorithm PPJoin [159]. The evaluation of the approach focused on time-performances, which showed how the proposed method outperforms SILK [147]. However, the comparison was made on heuristics supervised evaluation that the combination of the name and population was sufficient to discern cities in geographical data sources.

Rong, Niu, Xiang, Wang, Yang and Yu in [128] propose a distance metric for Linked Data instances, used then to feed binary classifiers to perform entity matching. In particular the authors propose three different metrics exploiting pre-labeling of the attribute values based on syntactical analysis. The authors evaluated the proposed metric on the OAEI 2010 Instance Matching test dataset, showing improvements with respect to tools participating at the context. It is important to underline how the solution was anyway tested based on training the classifiers on the evaluation dataset itself.

Sleeman and Finin in [135] propose a machine learning method for linking FOAF profiles. The authors consider inverse functionality of properties of the FOAF ontology, and produce different types of matching distance for attributes (e.g. simple matching, partial matching, cross-property matching). The authors then generate the training set with 500 sample equally distributed on match and non-match classes, and rely on SVM classifier to take matching decision. The authors further distinguish between easy cases, and more complex cases in the evaluation. Intuitively, the complex cases should be used for training, so that the overall classification could improve.

Notice that similarly to information system, the problem of information integration on the Linked Data, known as object consolidation, is usually managed and solved in a centralized fashion, dealing, among others, with the problems of high scalability required to handle entity matching problem on Web. Furthermore, in the context of the Semantic Web and Linked Data, when two entities are discovered to be the same, their identifiers are stated to be equal, materializing an *owl:sameAs* statement causing several issues [74]. Among other problems, once the equivalence between identifiers is materialized, it becomes hard to evaluate its reliability. Indeed, the information used to establish equivalence might change in time, potentially invalidating the equivalence stated. Another problem analyzed in [50] is related to the possible inconsistencies generated by the definition of conflicting equivalences. Given the scale of the information space forming the Semantic Web, an automatic approach for the execution of object consolidation is the only viable solution. With this respect, a recent evaluation of existing methods was proposed in [100].

3.3 String Similarity Metrics

Duplicate detection typically relies on string comparison techniques. There are different types of techniques to perform string comparison. In the following we present a selection of character-based string similarity metrics:

Edit Distance (or Levenshtein) [102]: this metrics refer to the minimum number of editing operation required to transform one string into another. This technique is good for typographical error, but not for other type of mismatches. It is commonly used in databases and indexes as a cheap and effective string similarity metric.

Affine Gap Distance [151]: this techniques allows to compute distances where words are truncated, e.g. John R. Bool VS Johnathan Richard Bool. The techniques is based on the concepts of *open* and *extended* gap, where opening gaps are penalized with respect to extended gaps. This allows to give lower costs to missing part of text with respect to edit distance.

Smith-Waterman Distance [137]: this technique extends the Edit and Affine gap Distance in which mismatches in the beginning and the end of the string are weighted less than mismatches in the middle of the string. This way, strings like “prof. Paolo

Bouquet, University of Trento" and "Paolo Bouquet, prof." can match within a shorted distance.

Jaro distance [89]: it was conceived as as string comparison algorithm for first and last names. It computes string length, find common characters in the same position of the index, each non-matching character is a transposition. The score is then calculated computing the average of the sum of rations between overlapping chars of the matched strings and the ratio of the transpositions.

Jaro-Winkler distance [155]: extends the Jaro distance by giving different weights to first part of the matching strings. This follows the intuition that the beginning of names are more relevant for matching decision than the tail.

Q-Grams distance [98]: the q-grams are short substrings of length q. The intuition behind the usage of these substrings is that similarity of 2 strings can be computed based on the numbers of q-grams they have in common.

Needlman-Wunsh distance [116]: a string similarity metric conceived in the context of molecular biology to match protein sequences. Scores for matching characters are represented as a similarity matrix providing score for each of the pairwise similarities. One of the parameters of this distance metric is the *gap penalty*, that is used to penalize gaps of missing characters.

Character based similarity metrics are essential to capture typographical errors, however there are other sorts of problems that can happen when comparing attributes from different sources. A type of error is related to different conventions in representing the exact same values. For example, the combination of *name* and *surname*, that can appear in different variants. For these reason, different similarity metrics considering token level similarity were conceived. In the following we present a selection of them:

Euclidean distance [56] p. 94: computes the geometrical distance of the tokens, or substrings, composing a string. The matching is computed considering each of the token as a dimension, and computing the distance computing the square root of the sum of the squares of distance between each of the dimensions. Notice that token based approaches assume exact equality in token comparison.

Jaccard distance [56] p. 293: considers two strings as two sets of attributes, and computes the ratio between the intersection of the sets and the union of the sets. This measure does not take into consideration order of the words.

Monge-Elkan [114]: this similarity metric considers atomic strings as strings separated by punctuation, and computes the score as the ratio between the number of matching atomic strings and the average number of atomic strings in the compared strings.

Overlap Coefficient: is related to the Jaccard distance, but in this case the similarity metric is the ratio between the intersection of the sets of substrings composing each of the compared string with the cardinality of the smallest of the two.

There is then a hybrid approach, combining token and chars similarity named TagLink. The concept underlying TagLink is rooted in the work of Cohen and Kautz [46], but was also further developed in [33] and applied on bioinformatics data. Basically, a graphical model is built out of the strings, and the problem of the distance consists of optimizing the cost of equivalence of each of the comparison. The method is quite sophisticated and allows to provide a middle ground between token-based and char based approach as sequences as both can be modeled in the graph. Bilenko et al. in [19] and [18] propose to learn adaptive similarity metrics for duplicate detection. Namely, rather than rely on heuristics that can be employed out of the box, the author propose to learn using support vector machines how specific attribute types should be matched.

For the sake of completeness, here we present also a small selection of phonetic similarity metrics. This type of metric relies on a transposition of words into representation of their pronunciation or sound. The approach can be very useful to capture misspellings or different or syntactic variations of words (e.g. Theater vs Theatre). Phonetic metrics are language dependent.

Soundex [130]: this is the most common phonetic coding scheme, and it is based on the assignment of identical code digits to phonetically similar groups of consonants. This system works well for caucasian names, and captures large part of relevant spelling variation with respect to the discriminative power of the full words. However, it worsen performance on asiatic names as the discriminative power relies on vowel (related to vocals) sounds that are ignored by soundex.

New York State Identification and Intelligence System (NYSIIS) [141]: this technique applies phonetic coding scheme considering vowels and replacing consonants with other similar letter. The coding results in a purely alphabetic string that showed to be slightly more precise than Soundex.

Oxford Name Compression Algorithm (OX-LINK)[65]: a two stage technique designed to remove defects of soundex. Essentially, each string is first encoded according to something similar to NYSIIS and then soundex is applied.

Metaphone and *Double Metaphone*[119]: it is a Philips alternative to soundex that uses a larger set of consonants to better reflect many english and non-english sounds. Metaphone allows multiple encodings to support large variety of pronunciation.

For further paper presenting alternative string similarity metrics, please refer to [58] and [47].

Part II

Vision, Theories and Definitions

Chapter 4

The Knowledge-based Matching Vision

This chapter aims at outlining a knowledge-based method suitable to solve the entity matching problem in the open context of the Web. Solving entity matching problem in the context of the Web is particularly challenging due to the semantic and structural heterogeneity characterizing data on the Web. We started conceiving the solution proposed in this work following the intuition that, when requested, people seem to be capable of taking relatively accurate matching decision considering incomplete sets of information, and relying on different types of knowledge to properly evaluate the relevance of the attributes matching (or not matching) in two descriptions. Capturing and using this "know-how", or *matching knowledge*, to build an explicit knowledge base seems to be a promising lead to define a matching approach suitable for a reliable solution of the open matching problem. It is important to underline the fact that any knowledge-based solution to the problem of entity matching in principle would be incomplete. In fact, to be complete, a knowledge-based solution would require to capture complete knowledge about any entity in the world which is in principle extremely complicated, if not impossible. However, recent dynamics related to the development of the Web 2.0 show that community efforts can produce relevant amount of shared and quite reliable knowledge easing the incompleteness problem. This work does not deal in depth with this problem/opportunity. However, knowing that in principle community efforts can help in scaling up the human effort and improve/extend the knowledge base is sufficient to justify the attempt of building such type of solution as it would be, in principle, sustainable.

The proposed method consists in an advanced feature-based entity matching solution based on the seminal paper [138]. In particular, it extends the feature-based

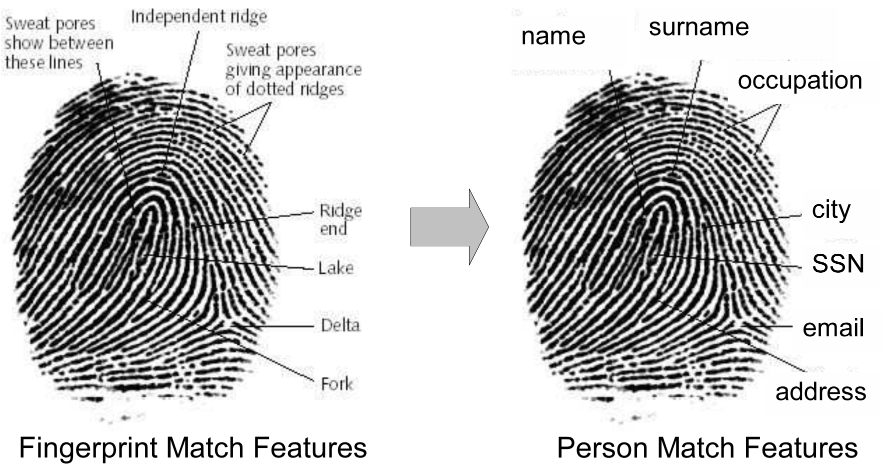

Sweat pores show between these lines
Independent ridge
Sweat pores giving appearance of dotted ridges
Ridge end
Lake
Delta
Fork
Fingerprint Match Features
name
surname
occupation
city
SSN
email
address
Person Match Features

Figure 4.1: Fingerprint Match Vision

approach proposed in [139], considering groups of features that must match to establish a positive entity match (i.e. matching rules). This approach is inspired by the methods adopted for fingerprint match analysis, and resembles the Inverse Functional Property Suite (IFPS) proposed in [118]. In fingerprint analysis, a matching decision is not taken based on one single feature of the fingerprint, but relying on combinations of biometric features, known as minutiae[1]. Each fingerprint is characterized by combinations of minutiae forming unique patterns [126]. If the pattern is found on two fingerprints, then a positive match decision is taken. In fingerprint analysis, the terms of comparison (i.e. features) used for fingerprint match are clear and known *a priori*, allowing the verification of the evidences that lead to a matching decision *ex-post*. These characteristics of the digital fingerprint matching process make its outcome reliable to identify people on a level to be used, or applied, in the investigation and establishment of facts, or evidences, in a court of law.

In a like manner, this work proposes to approach entity matching relying on a knowledge base formed by an *ontology* explicitly declaring the types of entities considered and the features type that must be considered along a matching process; and a set of positive or negative **identification rules** built on top of the ontology supporting and justifying any matching decision taken. We do not aim at defining a solution working on optimally matching entities in a closed environment, but rather we look at defining a solution that can be employed reliably out-of-the-box in an open context, and for

[1] Refer to `http://en.wikipedia.org/wiki/Minutiae` for an overview

example ease the problem related to unreliability of 'owl:sameAs' statements in the Linked Data [73].

From now on, we refer to the ontology underlying the proposed solution as the **identification ontology**, which has the twofold role of defining the terms considered relevant along a description comparison task, likewise fingerprint match analysis (see Fig. 4.1), and provide a central point of reference for the harmonization of the semantic heterogeneity characterizing entities' description on the Web. Indeed, once the terms of comparison are established *a priori*, it is possible to build and maintain, also with the support of automated tools for ontology and schema matching [59], a set of mappings between the known vocabularies and the identification ontology. A more in depth formalization and of the identification ontology is presented in section 5.

The **set of identification rules** built on top of the identification ontology aims at forming a general purpose equational theory, dictating the logic of matching as in rule-based entity matching proposed in [103, 79]. A positive identification rule resembles the concept of 'extended key' [103], as it aims at representing a combination of attributes identifying uniquely an entity in the integrated world (i.e. the Web). A more in depth formalization of the rules, and how we can obtain them is presented in section 6.

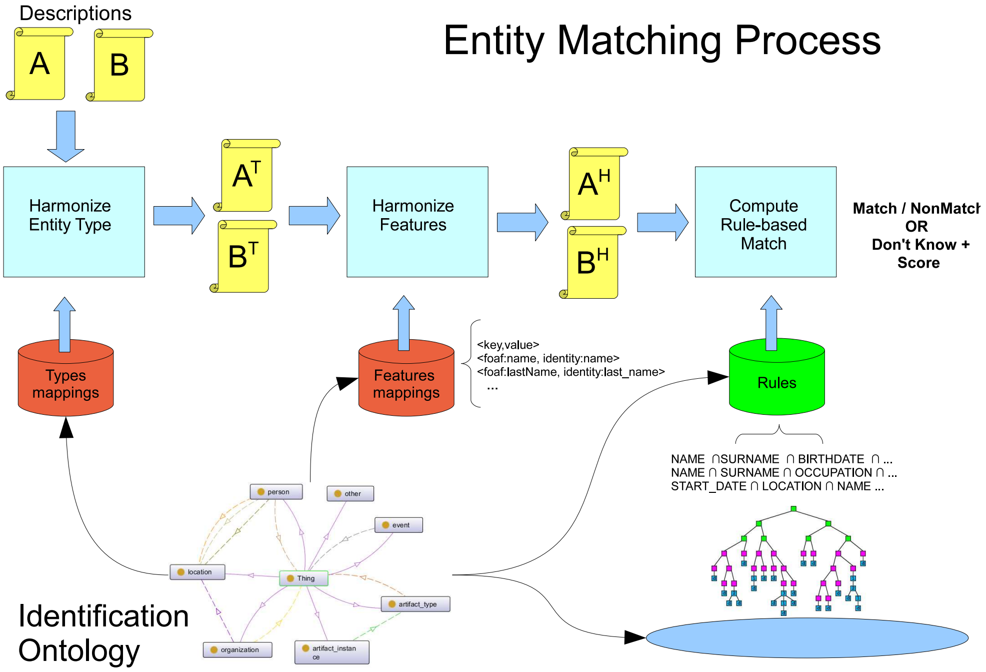


Figure 4.2: Knowledge-based Entity Matching Process

As analyzed in chapter 2, solving the entity matching problem in the context of the Web we are likely to deal with possibly underspecified descriptions when interpreted in the wide context of the Web. Hence, it becomes necessary to frame the solution under the Open World Assumption to accommodate cases where no reliable decision can be taken. Formally, the truth-value of a matching statement is independent of whether or not it is *known* to be correct by any single observer. From a practical point of view, the Open World Assumption implies that for the cases where positive matching conditions are not satisfied, it is not possible to automatically derive a negative matching conclusion. A more in depth analysis of the implication of the Open World Assumption on the application of entity matching rules is presented also in section 6.

Once we defined a sufficiently rich ontology presenting features supporting entity matching decisions, and a set of rules supporting reliable matching decision, we need to define a process capable of exploiting such resources to produce matching decisions. As a matter of facts, the set of rules defining the matching theory underlying this method are defined in terms of the identification ontology. Thereby, a necessary condition for the application of the rule is to harmonize the semantics of the compared descriptions towards the identification ontology. As previously mentioned, we rely on the existence of a set of semantic mappings supporting this task. A more in depth analysis and formalization of definition of these mappings is presented in section 5.6.

The most natural combination of all the different types of matching knowledge described so far is to put them in sequence. A graphical representation of the matching process is presented in figure 4.2. Namely, given a pair of descriptions A and B, these should first be harmonized based on the declared type as described in section 7.1. A pair of descriptions with harmonized entity type A^T and B^T now must pass a further harmonization step. In fact, we need now to harmonize the semantic of the attributes contained in the descriptions towards the one defined in the identification ontologies. In a sense, we need to spot the features that we know are essential to take matching decision, as described in section 7.2. Semantically harmonized descriptions A^H and B^H can now be compared, and matching decision can be taken relying on matching rules. The process of matching features and application of rules is described in section 9.2.

Chapter 5

Identification Ontology

In this section we define the Identification Ontology endorsing the semantics of OWL 2[1] for the definition of classes and properties. Namely in OWL 2 an ontology consists of 7-tuple $V = (V_C, V_{OP}, V_{DP}, V_I, V_{DT}, V_{LT}, V_{FA})$, where:

- V_C is a set of Classes containing at least top and bottom concepts *owl:Thing* and *owl:Nothing*;
- V_{OP} is a set of Object Properties containing at least *owl:topObjectProperty* and *owl:bottomObjectProperty*;
- V_{DP} is a set of Datatype Properties containing at least the data properties *owl:topDataProperty* and *owl:bottomDataProperty*;
- V_I is a set of Individuals (named and anonymous);
- V_{DT} is a set of Datatypes of D (i.e. *rdfs:Literal*) and other datatypes N_{DT};
- V_{LT} is a set of Literals LV^^DT for each datatype $DT \in N_{DT}$ and each lexical form $LV \in N_{LS}(DT)$.
- V_{FA} is a set of pairs (F, lt) for each constraining facet F, datatype $DT \in N_{DT}$ and literal $lt \in V_{LT}$ such that $(F, (LV, DT_1)^{LS}) \in N_{FS}(DT)$, where LV is the lexical fomr of lt and DT_1 is the datatype of lt;

The identification ontology will then present as classes the types of entities considered for the knowledge-based solution. Namely, every type considered will have to be declared as a Class. Following the notation of OWL 2, a class is declared as an entity identified by an Internationalized Resource Identifier (IRI)[2]. For example, the IRI

[1] http://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/

[2] http://www.ietf.org/rfc/rfc3987.txt

a:Person is declared as a class in the following way: $Declaration : Class(a : Person))$. Thus, the set of types T considered in the knowledge based solution consists of the classes contained in V: $T = \{V_C \setminus \{owl{:}Thing \cup owl{:}Nothing\}\}$. A knowledge-based solution can hardly be complete with respect to the heterogeneity and extremely wide variety of possible representation of the world. Thereby, it is necessary to draw boundaries to the domain represented in the ontology. The Okkam Conceptual Model (OCM) described in [30] provides an optimal starting point to define a model underlying the knowledge-based solution proposed in this work. Indeed, besides modeling the relation between entities and identifiers, the OCM embeds a simple taxonomy of top level, disjoint, categories. These categories are defined with the goal of providing an initial set of disjoint types covering large part of interesting entities consistently with other top level ontologies such as DOLCE[3] [62], Yago[4] [140] and OpenCyc[5] [107] and presented in [8, 7]. Quite intuitively, the types considered are: Person, Organization, Location, Event, Artifact Type and Artifact Instance and a bulk category Other. The Okkam Conceptual Model was carefully designed, but no formal ontology method was ever applied to validate its soundness with respect to the the rigidity of the classes in the taxonomy and the possibility of defining formal Identity Criteria for each of them. Therefore, in the following sections we present the results of the analysis of the OCM according to the OntoClean methodology [70], with the goal of validating the backbone taxonomy and remove inconsistencies, if any. The details of the types defined and a precise contextualization of the interpretation of these types is presented in section 5.3 as the result of a formal analysis of the Okkam Conceptual Model described in section 5.1 and 5.2.

As mentioned in the introduction of this chapter, the identification ontology is conceived not only to describe what types of entities are considered, but also what features are considered relevant for the matching task for each of the considered types. Thereby, the identification ontology will present as OWL2 Datatype properties and Object properties the features for each of the types. In particular, the set of features F considered consists in the union of the sets of Datatype and Object properties contained defined in the Identification Ontology IdO: $F = \{V_{OP} \in IdO \cup V_{DP} \in IdO\}$. Furthermore, besides the properties associated to considered types, we also declare specific meta-properties of these properties that can be useful to support the matching purposes. As we will show in the section 5.4, it may become necessary to extend the set of meta-properties for properties defined in OWL 2 in order to accommodate the level of expressiveness required. We are aware that OWL 2 and its dialects were conceived and

[3] `http://www.loa.istc.cnr.it/DOLCE.html`

[4] `http://www.mpi-inf.mpg.de/yago-naga/yago/`

[5] `http://www.cyc.com/platform/opencyc`

designed to correct issues introduce in the formalization of the first Web Ontology Language, and most of its changes are related to improving the computational tractability of reasoning tasks. However, as supporting scalable automatic reasoning is not the goal of the identification ontology, we believe we can take the license to introduce some modification without feeling the need of exploring in depth the implications with respect to this task. The problem may be considered in future developments of the method, but in this work is not considered particularly. A detailed list of features used for three of the considered types, together with the results of the ontological analysis, is presented in section 5.5.

In section 5.1 a brief description of the current state of the OCM is presented. In section 5.2 an introduction to OntoClean and the description of the analysis of the taxonomy are presented. In section 5.2.2 we propose solutions to inconsistencies, and in section 5.3 the backbone of the Identification Ontology as a result of an evolution of the Okkam Conceptual Model is presented.

5.1 Defining a Conceptual Model

The way people defines and uses identifiers to refer to entities has been under (philosophical) investigation for a long time, among others [101, 105, 104, 68, 92]. The subtle nature of the problem led to no complete and shared solution, leaving the issue unsolved. At this regards, we are assisting to a phase of transition where criticism proposed by Kripke [101] to the widely adopted *theory of description* [129] are well received, but at the same time no alternative solution gained consensus. Recent developments in the context of the Semantic Web (see [12]) are fostering further investigation about particular aspects and practical issues related to the problem of "Identity and Reference", aiming to find solution to the problem known in this context as"identity crisis" [44, 94, 63, 64, 23]. The focus of the problem is in the way URIs [6], adopted as syntactical solution for the representation of "names", should be used to refer to entities, and how naming and reference through URIs should fit the architecture of the Web. With respect to this, two main complementary approaches emerged: (1) Linked Data approach [13], the Okkam approach [28].

The Linked Data initiative, promoted by the W3C, endorses the approach supporting the fact that names for entities are equivalent to description. Thereby, in this context the creation and definition of URIs must always be associated to a description of the entity identified. This solution has appreciable practical advantages as it fits the current architecture of the Internet. In fact, the supporters of the Linked Data

[6] Uniform Resource Identifiers, RFC3986

initiative suggest to adopt URL[7] as identifiers for entities [20]. This solution has the advantage of defining globally unique identifier that are also de-referenceable into documents containing a description for the entity [22]. This mechanism allows the definition of the Web of Data, relying on existing infrastructures and protocols defined for the Web. The main drawbacks of this approach are two: first, there is no lookup system to find and reuse names referring to entities and this has the side effect of causing a proliferation of names limiting effective data integration; second, the equivalence between URIs managed through "OWL sameAs" statements is disputable and requires the computation of transitive closure along the Global Giant Graph [14].

Okkam was a large scale integration project co-founded by the European Commission aiming at the definition and development of an Entity Name System (ENS) as backbone service handling the process of assigning and managing the lifecycle of globally unique identifiers for entities in the WWW [28]. These identifiers are global, with the scope of persistently identifying entities across system boundaries. The ENS has a distributed repository for storing entity descriptions and their identifiers. An entity profile is essentially a relatively small amount of information on each entity identified. Clients can interact with the ENS and may inquire for entities' identifier by providing keywords about those entities. If the entity does not have and identifier in the ENS, a client can create a new one in which case the ENS returns the newly assigned identifier. The result of a consistent adoption of the 'OKKAM' approach is that all resources presenting references to the same entity are assigned the same identifier. The Okkam approach can be seen as orthogonal and complementary with the Linked Data one. In order to promote the role of the ENS as means for smooth and frictionless data integration, entity profiles present Web of Data URIs as alternative identifiers for Okkam entities they co-refer to. More information about the ENS can be found in [27, 26].

To make explicit the role of the Entity Name System as a solution to the identity crisis, a conceptual model describing the domain of entities and their relation with URIs as their identifier on the Semantic Web was defined [30]. This model, named Okkam Conceptual Model (OCM[8]) was first defined to provide an explicit representation of domain in which the ENS was proposed as infrastructural element, and to support and justify technical solution implemented within the ENS.

The conceptual model for Okkam is aimed at providing an explicit representation of "the world of entities and their identifiers", extending and adapting the model proposed in [63, 64]. In particular the model describes how entities are represented and identified

[7] Uniform Resource Locators, RFC1738

[8] An OWL implementation of the OCM: `http://models.okkam.org/ENS-meta-core-vocabulary.owl`

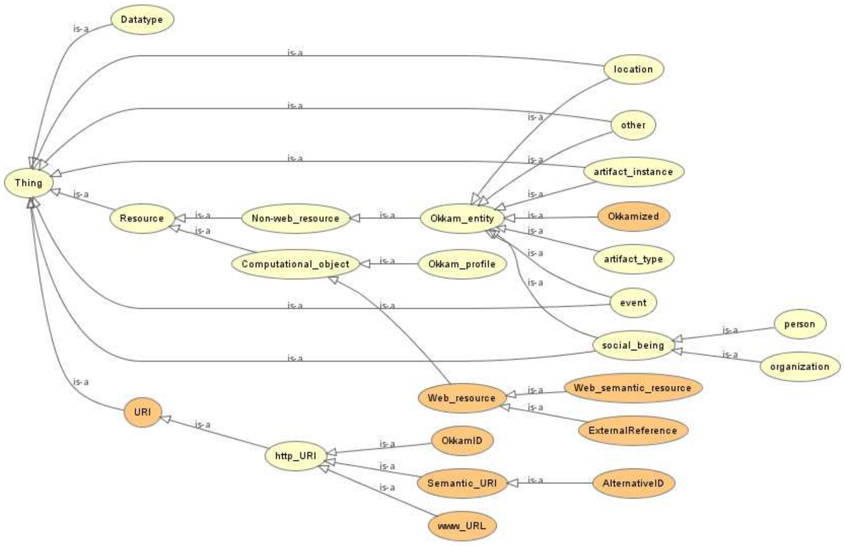


Figure 5.1: The taxonomy backbone of the Okkam Conceptual Model

in the context of the (Semantic) Web. In this context, the founding pillars are the concepts of Resources and URI. The definitions of Resources and URI (RFC 3986[9]) are circular, and make the two things dependent from each other. Indeed, essentially the definitions say that "*a Resource is anything that can be identified by a URI*" and "*a URI is an identifier for a Resource*". These two types are then dissected into sub classes, defining more fine grained classes to represent specific types of Resources and URIs. A graphical view of the backbone taxonomy of the Okkam Conceptual Model is presented in figure 5.1. A graphical view of the whole conceptual model including relations between concepts (object properties) is presented in figure 5.2.

An important part of the conceptual model is the definition of the taxonomy of types supported by the Entity Name System. Given the definition of OkkamEntity as a subclass of Non Web Resource, it was necessary to define several macro categories dividing the domain of entities treated in the Entity Name System for practical reasons. In fact, such categories are used to support the user in providing a first very coarse-grained disambiguation about the entities searched/created. Furthermore, knowing these general types for entities enables the possibility of defining specific 'matching methods' given the typical set of information used to identify types of entities. For example, knowing that an entity is of type Person allows the application of matching heuristic capable of finding more effectively the description matching a query. With this goal, the Okkam Conceptual Model defined a simple taxonomy presenting concepts as Location, Person, Organization, Event, Artifact Type and Ar-

[9] http://www.ietf.org/rfc/rfc3986.txt

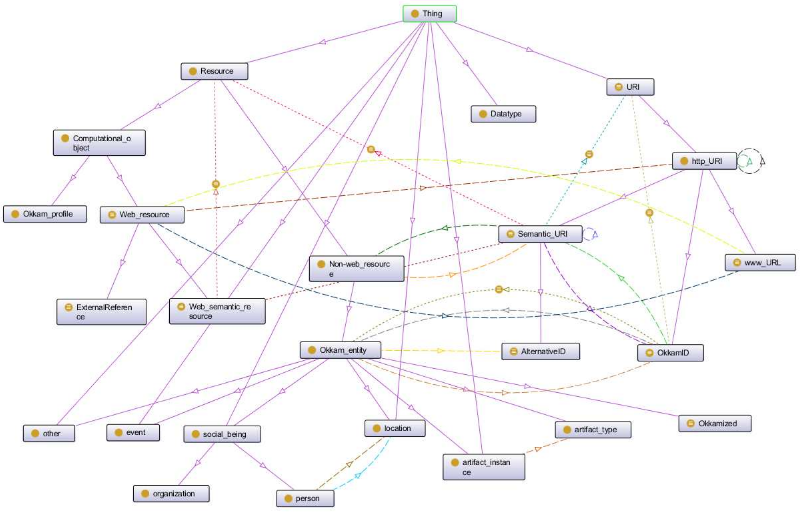


Figure 5.2: Okkam Conceptual Model: concepts and relations

tifact Instance and a bulk concept named Other. It is important to notice that the categories represented in this model are common to most of the knowledge bases available online. In fact, this list is not the mere result of an intuitive enumeration, but is the result of an analysis of the most popular and accepted models in the community [7].

5.2 Formal Analysis of the Conceptual Model

OntoClean is a formal methodology aimed at guiding ontological investigation with the goal of defining valid taxonomic relations between concepts [68]. The methodology is based on ontological notions taken from philosophy such as *Essence*, *Identity* and *Unity*. These are used to define formal meta-properties characterizing concepts (i.e. properties). Such meta-properties are then used to define constraints for the definition of correct subsumption relations. In particular the notions considered by OntoClean are:

- Rigidity (R): a property is rigid if it is essential to all its possible instances which cannot stop being instances of that property in all possible world they exists (e.g. being a human being). With respect to rigidity, a property can be **rigid**(+R), **semi-rigid** (-R) or **anti-rigid** (∼R). Semi-rigid properties are essential to some instances, but not to others (e.g. being hard). Anti-rigid properties are not essential at all for all their instances (e.g. being a student).
- Identity (I): this notion refers to the capability of recognizing individuals to be the

same. Namely, a property has identity criterion when it is possible to understand whether entities satisfying that property can be compared to be identical. Identity criteria are used to determine equality between entities. There are two types of identity criteria, synchronic and diachronic. Synchronic identity criteria are those that allow to recognize entities at a specific time. Diachronic identity criteria are those that allow to recognize an entity along time. OntoClean proposes to characterize properties analyzing whether these **carry inherited** (+I), **supply** (+O) or **do not carry at all** (-I) identity criteria. A property carries inherited identity criteria when it is possible to determine equality between entities on the base of qualities that are inherited from more general properties in the taxonomy. A property supplies identity criteria when qualities specific of the entities satisfying this property can be used to determine equality between entities.

- Unity (U): this notion refers to the capabilities of recognizing the boundaries and the part of an entity, such that it is possible to understand whether these entities exist as whole. The unity criteria (UC) are determined by unifying relations that can be used to define a kind as whole. Example of unifying relation can be used to distinguish topological wholes (e.g. piece of stone), morphological wholes (e.g. constellation) or functional whole (e.g. hammer). OntoClean proposes to characterize the properties with three meta-properties, **unity**(+U), **no unity** (-U) and anti-unity (∼U). A property has unity meta-property when it has common unity criteria among all instances (e.g. a person). A property has no unity meta-property when does not have uniform unity criteria among all instances(e.g. legal agent). Finally, a property has anti-unity meta-property when does not have any unity criteria at all (e.g. amount of water), and thus it is not possible to find a unifying relation defining a whole.

The notions above are then used to define taxonomic constraints that must be respected to create valid subsumption relations between concepts. Such constraints are:

1. anti-rigid properties cannot subsume rigid properties (+R $\not\subset$∼R);
2. properties with identity criteria cannot subsume properties without identity criteria (-I $\not\subset$ +I);
3. properties with unity criteria cannot subsume properties with no unity criteria (-U $\not\subset$ +U);
4. properties with unity criteria cannot subsume properties with anti-unity criteria (+U $\not\subset$∼U)

The core part of the OntoClean methodology is:

1. to analyze the taxonomy and assign the aforementioned meta-properties to all the concepts of the taxonomy;

2. check whether taxonomic constraints are respected;

3. modify the taxonomy to respect the constraints;

For more information about OntoClean, refer to [68].

5.2.1 OntoClean Meta-properties Annotation

The first important step of the OntoClean methodology is to analyze the backbone taxonomy and label each property (or concept) with specific marks representing its evaluation with respect to Rigidity (R), Identity (I) and Unity (U). In the following, each paragraph presents a description summarizing the analysis leading to the assignment of each metaproperty.

Thing OWL:THING is the most basic concept of an ontology. Everything is necessarily a "*Thing*" and does not stop being a "*Thing*" along its existence, thus being a thing is a rigid property **(+R)**. By definition a *Thing* does not have defined criteria of identity **(-I)** and unity **(-U)** properties.

Resource According to standard the definition [10], a resource "*is anything that can be identified by a URI*". It is important to notice that this definition is circular with respect of what a URI is. However, if something can be identified, it is possible to define sufficient own identity criteria. Thereby, a resource supplies identity criteria **(+O)**. The concept of RESOURCE is so broad and it involves potentially any other concept with different unity criteria. For this reason, this concept has no-unity property **(-U)**. Intuitively, *being a Resource* is an essential property of all resources so, RESOURCE has rigidity property **(+R)**.

Computational Object A COMPUTATIONAL OBJECT is *the physical realization of an information object and something that might participate in a computational process that ensures the resolution of a URI.* Computational object in its generic conception is a rigidity property **(+R)**, has clear identity and unity criteria as each of them present finite physical realization that can be compared to determine equality **(+O)**, and they

[10] Definition of URI in RFC3986, `http://tools.ietf.org/html/rfc3986`

can be considered wholes in the sequence of representing them on some type of physical support (i.e. magnetic hard-disk) **(+U)**.

Non-web resource A Non-Web Resource is "*the class of resources that are not computational objects*". This definition is quite broad, and essentially it includes anything but computational objects. A Non-Web Resource inherits identity criteria from Resource **(+I)**. *Not being a computational object* is a rigid property of Non-Web Resource **(+R)** and does not present clear unity criteria **(-U)**.

Web resource A Web Resource is "*the class of computational objects accessible on the Web by dereferencing a URL.*". A web resource, as defined, inherits identity (+**I**) and unity properties as they can be defined as wholes in the sequence of bits composing their physical realization **(+U)**. A web resource is a computational object that is accessible on the Internet, but this fact does not appear to be a rigid property of the object itself. For this reason, *being a web resource* does not seem to be a rigid property of a computational object and thus it has anti-rigidity property **(∼R)**.

Okkam profile An Okkam Profile is "*the set of information about an Okkam entity stored at the Okkam repository*". According to this definition, an Okkam profile is constituted by a set of information about an entity that might change along time. The main property of an Okkam Profile is to be about an Okkam Entity, and this can be seen as a rigid property of the Computational Objects that are Okkam Profiles. Indeed, *being an Okkam Profile* implies its existence within the Entity Name System and thereby, by definition, this Okkam Profile must be about an Okkam Entity. It seems reasonable to conclude that *being an Okkam Profile* is a rigid property **(+R)**. An Okkam Profile has unity criteria **(+U)** as any Computational Object and inherits identity criteria **(+I)**.

Okkam entity An Okkam Entity is "*the class of resources that can be given an OkkamID. It includes only entities that have spatio-temporal or at least temporal properties*". The property of being an Okkam Entity refers to all particulars and concrete entities, including events but excluding all abstract entities and concepts. Intuitively, being an Okkam Entity is a rigid property of all its instances **(+R)**, inherits identity criteria from Resource **(+I)** and has no clear unity criteria as the varieties of possible instances makes impossible to imagine common unifying relations defining wholes **(-U)**.

External reference An EXTERNAL REFERENCE is "*the class of web resources that give information about resources and is disjoint from the class of Web semantic resource*". According to the definition, an external reference is a Web Resource providing further information about a resource without being a Web Semantic Resource. Being an External Reference does not appear to be neither essential nor rigid property of a Web Resource, so the this property is anti-rigid (∼**R**). On the other side, External references inherits both unity (+**U**) and identity criteria (+**I**) from Web Resources.

Web Semantic Resource A WEB SEMANTIC RESOURCE is "*the class of web resources that are accessible by dereferencing a semantic URI which redirects to another WWW URL which gives access to the web resource*". Essentially, Web Semantic Resources are those Web Resources forming the Web of Data promoted by the Linked Data initiative. The property of *being a Web Semantic Resource* does not appear to be rigid with respect to the computational object realizing them. A Web Semantic Resource is a computational object representing a set of semantically annotated information that is accessible on the Web. Thus, as for Web Resources, also Web Semantic Resources are anti-rigid (∼**R**). Web Semantic Resource property inherits identity criteria from (+**I**) Web Resources, and present clear unity property as any computational object (+**U**).

Social Being A SOCIAL BEING is defined as the class of "*intelligent agents whose status as an agent is acknowledged within some social system, and who is capable of playing certain social roles within that system*". According to the definition, *being a social being* is anti rigid property (∼**R**) of as a "status acknowledged within a social system" can be interpreted as an agent having some role recognized by in a social system. Thus, assuming that we could remove a whole social system apart from one of its element, for example due to a terrible disease or a nuclear disaster, this element would exists *without being a social being* anymore as there would not be a society acknowledging it. *Being acknowledged by a society* does not allow to define commonly shared identity criteria (**-I**) nor define precise unity condition (**-U**). It is important to notice this problem is due to the fact that social being is quite a vague concept.

Person The property PERSON has been widely analyzed in other works such as [70], thus referring to that work can easily realize that the property of *being a person* is rigid with respect to all its instances, and that this property has clear unity (+**U**) and its own identity criteria (+**O**).

Organization Also the property ORGANIZATION has been analyzed in [70], and following the intuition of the authors we can say that being an organization is a rigid property (+**R**), presenting identity (+**O**) and unity criteria (+**U**). Finding a common unity criteria defining an organization as whole seems to be possible relying on the functional notion of unity. Namely, an organization can be seen as a whole as it functions as whole, not only because of the mereological sum the its part.

Location A LOCATION is defined as "*the class of geographical and physical locations (e.g. London, Canada, Africa, S. Peter square)*". The property of being a location is rigid with respect to all its instances **(+R)**. A location seems to have its own identity criteria (+**O**) as, intuitively, it seems to be possible to determine equality between two location by analyzing their mereological extension [68]. However, this type of criteria seems to be suitable only to establish synchronic identity, but it would create problems along time. From an ontological perspective, it seems that a location does not have a clear unity criteria, indeed it is impossible to establish uniform unifying relation between the parts of a location (∼U) to form a whole without depending on unity criteria of other entities. This problem is due to vagueness, and intuitively one may think that it could be solved by 'convention'. Namely, it should be possible to draw arbitrary limits to location and if those limits are shared than one could talk about location as whole. However, it seems that a unity criteria for location based on arbitrarily assigned overlays would make the unity of a location dependent from the unity criteria of the overlay, and thus we conclude that location has anti-unity property.

Event The concept of EVENT is defined as "*the class of event, including natural events (e.g. hurricane, earthquake, etc.), social events (e.g conference, meeting, wedding, etc.), economical events (e.g. closing deals, signing agreements, merging and acquisitions, etc)*". Events are entities under the lenses of philosophers for many years, but no shared agreement over their nature was ever found. In particular, it seems hard to find a common and satisfying solution with regards to the identity and the unity criteria of an event. Despite this, an analysis of some literature was performed with the aim of finding some evidences of approaches that could be applied dealing with 'events' in the context of the Okkam project. The first paper considered was a famous work of Davidson [51]. In this paper the author attempts to give a definition of events on base of 'cause and effects'. Namely, if two events have the same cause and the same effects then they must be the same event. In particular, the identity criterion would be: $(x = y\ if\ and\ only\ if\ \ ((z)(z\ caused\ x \longleftrightarrow z\ caused\ y)\ and(z)(x\ caused\ z \longleftrightarrow y\ caused$

$z)))$. This would answer the question about identity criteria for events. Unfortunately, it was pointed out that the identity criteria given by Davidson suffers from some kind of circularity, as pointed out by Quine in [121]. Indeed, this type of definition, relying on events (i.e. causes) to identify events relies on the assumption that events are actually individuated. Thereby, the identity criteria given by Davidson is not satisfying. However, if we consider the pragmatic approach proposed by Kellenberg in [92], the definition of identity criteria were often confused with the definition of identification criteria. In particular Kellenberg criticizes the approach of Lowe [105, 104], and proposed a different one. Getting a little into detail, Lowe's identity criteria follows the canonical form: $(x)(y)(\Phi x \wedge \Phi x) \rightarrow (x = y \leftrightarrow Rxy)$. So, if the variables x and y are Φ's it implies that x and y are identical if and only if it exists x and y stand with respect to a functions R (capable of establishing their identity). It is important that the property Φ is not considered in the definition of R, to avoid the circularity problem affecting for example the identity criteria for events defined by Davidson. Kellenberg highlights the fact that this identity criteria presents itself some circularity, indeed it presupposes the capability of identifying and discerning x and y: "*how can we know which one that relation is, and what Rxy means unless we know already what it means to be a single* Φ *and, hence, unless we already know the criterion of identity for* Φ*?*". What Kellenberg proposes is a pragmatic definition of identity criteria as "*doing that, if successfully performed, pick out only single entities of the kind in question from all single and all pairs of entities relevant in the domain*". Taking in consideration the criterion of identity proposed by Davidson, the criterion consists in verifying with regard to an entity (x, y) of an unspecified domain D that x is an event, that y is an event and that x and y have the same causes and effects. This pragmatic conception of criteria of identity shows identity criteria as concrete entities as they must be performed on particular entities at particular times and in possible worlds. Kellenberg further states that to understand whether an entity is part of a class, we need to define the class, and defining the class we have the criteria of identity for the members of the class. In order to understand whether two entities are the same, Kellenberg proposes the definition of an 'individuator', which is another doing entailed by the identity criteria. As identity criteria are *doings*, they don't have logical from, but sentences expressing identification of such criteria have. In particular, the *logical standard form of a criterion of identity* (InS) would be: $(x)(y)(I_A xy \leftrightarrow E_A x \wedge E_A y \wedge J_A xy)$, where x and y range over the entities to which the expression denoting the identity criteria I_A can be applied, E_A is a complex predicate denoting the identity criteria for As and J_A is a complex predicate denoting the individuator for As. Other articles were considered pursuing for identity criteria for events. Despite admitting to be unsuccessful in the individuation of events, Unwin [145] proposes a schema in which sortal terms should fit to be recognized as events, and further suggests how events and facts seems to be strongly correlated. Cleland [45] proposes to use changes (or concrete phases) as basic individuals for the identification of events. According to Cleland, *concrete phases are enduring and unrepeatable denizens of physical reality* without extending in space, thus concrete phase can co-occur in the same place. This article is very interesting and promising, but a deeper discussion of its proposal is beyond the aim of this work. Finally, Carlson [34] proposes a linguistic analysis where thematic roles would have a conceptual role in the individuation of events starting from the principle that 'an event has at most one entity playing a given thematic role'. The literature analysis performed does not have the ambition to be either complete or sufficient to state that identity and unity criteria for events in its more general conception exist. Nevertheless, given the pragmatic approach to the definition of identity criteria proposed by Kellenberg, we could say that identity criteria for events exists, and are explicitly the one used to define events in the context of the Entity Name System (**+I**). Furthermore, one could argue that 'concrete phases' could be proper unity criteria for events. Indeed they seem suitable to represent events co-occurring in the same spatial region and at the same time. The sum of concrete-phases occurring in a spatial region and along a temporal interval could be the unity measures to define complex events, and thus we would be tempted to say that events do have unity criteria (**+U**). Furthermore, if we consider the sum of concrete-phases as being the cause of some specific state, thus having a precise effect on the world, we can see also an event as whole from a 'cause-effect' or functional perspective.

Artifact Type An ARTIFACT TYPE is defined as "*the types (or models) of an artifact which are used to produce an arbitrary number of copies (artifact instances). Examples are: Opel Zafira 2.0 DTI version 1, MS Word 2007, the Othello by Shakespeare. Notice that the class of bridges is not the artifact type of the London Bridge, as the concept of bridge is not the model from which copies are produced. Works of arts that can be reproduced in copies are members of this class*". Being an artifact type seems to be a rigid property for a model. One could argue that once the model is 'out of production', or never gets to production, a model stops to be an artifact type as there are no instances. Nevertheless, once an object is defined as an artifact type, potentially it can be used at any time to be a model for the creation of instances of that artifact type. Thus, we can say that artifact type is a rigid property (**+R**)(similar conclusion are also in [35]). Artifact types have the property of being unique, or being the unique models for an arbitrary number of instantiation copies. Intuitively, it seems then that artifact types have their own intrinsic diachronic and synchronic identity criteria as two different models will be realized in different types of artifacts. Adopting a pragmatic approach as proposed by Kellenberg in [92], seems clear that identity criteria exists, indeed intuitively people have no problem to individuate, the artifact type of artifact instances, both distinguishing different types and recognizing when two artifacts are of the same type. Thus, it seems reasonable to conclude that artifact types do have identity (+**O**) and unity criteria (+**U**). However, artifacts are still under the lens of philosophers (e.g. [36])and works as [35] ended up with different conclusions.

Artifact Instance An ARTIFACT INSTANCE is defined as "*the class of concrete artifacts, like the London Bridge, my own Opel Zafira, my copy of the Othello, my installation of MS Word, etc.. Not to be confused with the class of artifacts-type, like Opel Zafira, Shakespeare's Othello, MS Word*". As for ARTIFACT TYPES, being an artifact is a rigid property of every artifacts. The non-rigid property of the artifact is the role it plays with respect of its function, but every artifact does not stop being such until it gets destroyed (+**R**). Finding a 'metaphysic' identity criteria shared among all artifact seems to not be feasible. For sure, all artifacts have the property of being somehow the result of a process driven by human activities. However, a possible identity criteria could be the fact that artifacts can said to be the same if the they are composed of exactly the same components and exactly by the same matter (+**I**). This same vision is shared also by the authors of OntoClean methodology, despite someone does not agree [35]. Artifact instances have intuitively clear unity criteria, indeed where identity is tricky, people usually don't have problems in discerning different artifacts of the same type. However, there are types of artifacts (e.g. wine) that do not have clear unity property, if not depending from others. Thus it seems reasonable to state that not all artifact instances present unity criteria (**-U**).

Okkamized OKKAMIZED is defined as "*the class of Okkam entities that are assigned an OkkamID*". Being okkamized is not an essential properties of all Okkam entities. Indeed, it might happen that an Okkam entity exists without being assigned any OkkamID, so being okkamized is a anti-rigid property (∼**R**). Okkamized entities inherits identity criteria from Okkam entities (+**I**) and does not seem plausible to find a common unity criteria for all okkamized entities (**-U**).

Other The property OTHER defines "*the class of things which do not fall under any other predefined class of OKKAM entities*". This property has to be intended as a utility to represent all those entities that cannot be classified according to the main set of class analyzed. Nevertheless, being a other is not a rigid property (∼**R**), and does not define identity (**-I**) and unity criteria (**-U**).

URI The definition of URI (Uniform Resource Identifier) is given in RFC3986, `http://tools.ietf.org/html/rfc3986`, "*A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies*[11] *an abstract or physical resource. The URI syntax defines a grammar that is a superset of all valid URIs, allowing an implementation to parse the common components of a URI reference without knowing the scheme-specific requirements of every possible identifier*". According to this definition, a URI has its own identity criteria as explicit identity (+O) property are defined and proper of all URIs. The combination syntax, encoding, size constrains seems to suggest that a URI has also unity criteria (+U). According to the definition of URI, a URI is such only if it is associated as identifier of a resource. This fact seems to suggest that *being a URI* is not an essential property of all the strings respecting the URI syntax, and thus the property of being a URI is anti-rigid (∼R). Indeed, at some extent, the definition of URI seems describing a role with respect to the Resource it is attached to. Intuitively, it is impossible to understand if a URI *per se* identifies any resource at all. The only rigid property of a URI is the syntactic one, so at worst we could define the class of "URI-like" strings as rigid.

HTTP URI HTTP URI is defined as "*the class of de-referenceable URIs*". The definition makes implicit reference to the HTTP protocol as mean to dereference a resource identified by a URI. Similarly to URI, the property of *being an HTTP URI* has unity (+**U**) and identity properties (+**I**). Also in this case, intuitively an HTTP URI does not guarantee *per se* that there is a resource de-referenceable related to it. Thereby, being an HTTP URI is an anti-rigid property (∼**R**).

OkkamID OkkamID is defined "the class of OkkamIDs", namely those URIs that are assigned to Okkam Entities in the the context of the Entity Name System. Similarly to URI, the property of *being an OkkamID* has unity (+**U**) and identity properties (+**I**). As to rigidity the class of OkkamId requires a more detailed analysis. First of all, any OkkamId is generated just and only when a new Okkam Entity is Okkamized, namely it is assigned an OkkamId. This means that since its generation, and OkkamId is necessarily tied to an Okkam Entity. Furthermore, in principle no Okkam Entity should be removed by the Entity Name System, and thus for all its existence, even

[11] We believe it would be more correct to say that a URI is used to identify resources, rather than identifies. Indeed a URI is not a property a resource, but rather a label/name stick to it for sake of making the reference non ambiguous.

along the lifecycle of the Okkam Entity, and Okkam Id is rigidly tied to an Okkam Entity. This principles of persistence and consistency gives, in my opinion, rigidity property to an Okkam ID. Namely, every OkkamID is necessarily tied to the Okkam Entity for all its existence. An OkkamId does not have merely the role of dereferencing objects, but to be a rigid and shared name for entities. Thus, according to my intuition I affirm that being an OkkamId is a rigid property (+**R**).

Semantic URI A Semantic URI is defined as "*the class of URI that are identifiers of resources. They refer to resources by description since when they are dereferenced they redirect to other URIs which resolve in web resources that give description of the referred resources*". Similarly to URI, the property of *being an Semantic URI* has unity (+**U**) and identity properties (+**I**). The automatic de-referentiation mechanism underlying the existence of Semantic URIs suffers from the same lack of rigidity as regular HTTP URI. Indeed, in principle a Semantic URI does not guarantee *per se* that there is a Semantic Resource de-referenceable related to it. Thereby, being a Semantic URI is an anti-rigid property (~**R**). Similar analysis can be done for the class of WWW URL.

AlternativeID An Alternative Id is defined as "*the class of semantic URIs which stand in the co-refer relation to OkkamIDs*". Namely, an Alternative Id is a URI which is discovered to identify and refer an okkamized Okkam Entity. As Alternative Id is dereferenced outside the Entity Name System, and thus in a context that does not guarantee persistence and consistency of the reference, is not a rigid property. Nevertheless, as the other URIs the property of *being an An Alternative* has unity (+**U**) and inherits identity properties (+**I**).

5.2.2 Constraints Violation in Backbone Taxonomy

A view of the taxonomy labeled according to the analysis described in section 5.2.1 is presented in Figure 5.3. An analysis of this taxonomy with respect to the metaproperties assigned helps highlighting problems or misconception in the definition of the model.

The first constraints check is about the rigidity meta-property. As shown in Figure 5.3, many properties included in the model as classes are anti-rigid. These classes anyway are clearly part of the domain the Okkam Conceptual Model wants to represent. A deeper analysis, presented in section 5.2.2, will highlight that most of them can be actually better represented as roles. In particular, the properties presenting constraints violation to be removed from the taxonomy are: Social Being and OkkamId. Both the

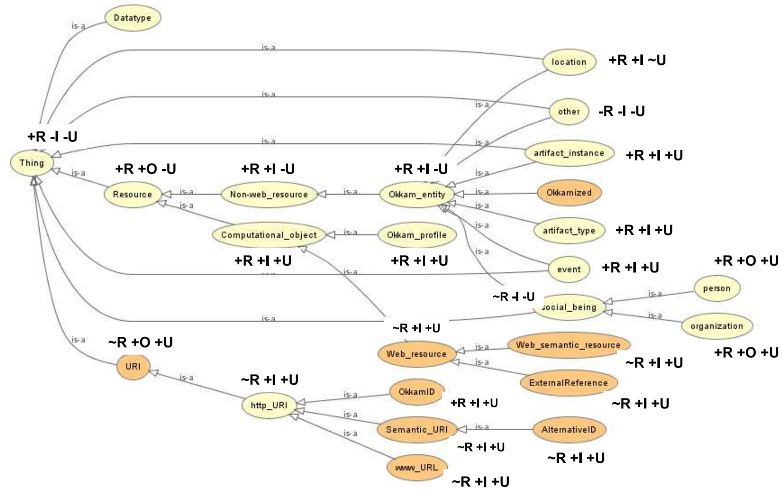


Figure 5.3: Okkam Conceptual Model Labeled with OntoClean metaproperties

properties break the OntoClean constraint (+R $\not\subset$~R). Thereby, the relation between Social Being and properties Person and Organization should not be represented by subsumption, but should be represented by a disjunction. That is, if something is a Social Being, then it is either a Person or an Organization. Analyzing the taxonomy it seems clear the intent of finding a common ancestor for person and organization as social agents. However, I believe that a better way to model this would be to model legal agent as a role, and state that persons and organizations play the role of social agents. Regarding OkkamId property, it seems that this cannot be in subsumption relation with URI, but rather it is constituted by a URI and inherits its syntactic characteristics. Anyway, as we concluded that an OkkamId is rigid property with respect to its property of being about an Okkam Entity, then we have to remove OkkamId from the subsumption relation with the URI.

Another subsumption that violate an OntoClean constraint is the one between Okkam Entity and Other. This property is too vague to be maintained in the taxonomy and subsumption relation with Okkam Entity violates the (-I $\not\subset$ +I) constraint. Okkam Entity has identity property, but the property Other seems not to have it, and incompatible identity criteria are sign that the properties are disjoint.

The taxonomy defined by rigid classes does not present particular inconsistencies with the constraints defined by the OntoClean methodology.

According to the methodology, the first stage after considering rigidity, is to individuate the so called phased sortal. A phased sortal is a property that changes identity criteria along its existence, while remaining the same entity. Analyzing the taxonomy, potential candidates for being phased sortal could be Web Resources, URI and their

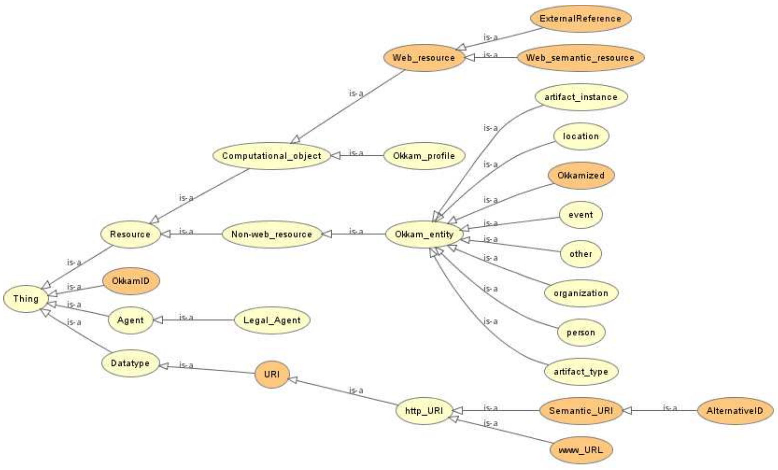


Figure 5.4: Okkam Conceptual Model taxonomy modified to remove constraint violation.

subclasses. Nevertheless, these classes don't change their identity criteria along their existence and thus, we can conclude that there are no phased sortal in the taxonomy.

The next step is analyze the presence of roles. Performing this task we can identify a set of classes that seems to play some role, in particular: Web Resource, Web Semantic Resource, External Reference. All these classes seems to share the same kind of problem with respect to rigidity. In particular all of these are computational objects that happen to be accessible on the Web, and thus depend on two things: the host server and the URI that is resolved in the specific location of the host. Without these two essential elements, no computational objects can play the role of being a web resource of any type. The property URI and its subclasses do not seem to play a role. One could think that they play the role of being the identifiers of resources, but in my opinion this is not a role. URIs are properties following specific syntactic constraints that can be employed in the system of the web to identify/locate resources. Their existence is independent from the resources they refer to. Indeed, nobody would deny that a string representing a URL is not a URL despite no computational object can be retrieved resolving it according to any of the HTTP protocols. In my opinion, URIs are particular data types, more specifically strings that respect a defined grammar (URI), declare specific protocols for its resolution (HTTP URI), follow specific conventions (WWW URL, Semantic URI), or present some common pattern (OkkamID). A view of the taxonomy modified to remove constraint violation is presented in Figure 5.4.

An Evolution of the Model

In order to fix the inconsistencies of the Conceptual Model for Okkam aimed at representing the context of naming and reference in the (Semantic) Web, I propose the following modification of the model:

- define a role hierarchy for the properties Web Resources, External References, Web Semantic Resources;
- define properties representing the abstract locations that can be referenced by URI and where computational object can be placed;
- define a role hierarchy for agents and create the property of Legal Agents so that Organization and Person can be referred
- define URI and its hierarchy as subclass of Data Type, in particular the properties should identify strings presenting particular syntactic properties (e.g. URI-Like Strings). These novel data types then should be used as range of properties related to Web Resources.
- explicit the interpretation of the Location concept to be interpreted more specifically as Geo/Political Feature, as these are the type of entities of interests in the context of the Entity Name System. Indeed, the class of location is aimed at identifying the features such as cities, states, roads, building presenting particular geographical and spatial properties providing identity and unity properties. As showed in the analysis presented in section 5.2.1, locations are quire slippery to identify *per se*.

Furthermore, it would be interesting to consider other properties aiming at representing further details of domain represented. An interesting feature would be to represent the resolution mechanism as the combination of protocols and the agents involved in the process of retrieving a computational object given a URI. A further aspect to explore would be an analysis over the representation of authorities behind the resolution mechanism. In order to complete the model, it would be interesting to find a way to represent the difference between the description provided by a Web Semantic computational object playing the role as Web Semantic Resource and the description embedded in the computational object playing the role as Okkam Profile. In particular, it would be important to highlight the non-authoritative/open nature of the description provided by Okkam Profiles in comparison with the Web Semantic Resources. However, these aspects are not relevant for the main goal of this work.

5.3 Identification Ontology Taxonomy

In previous sections an analysis over the Okkam Conceptual Model was presented. In particular, the conceptual model was dissected according to the OntoClean methodol-

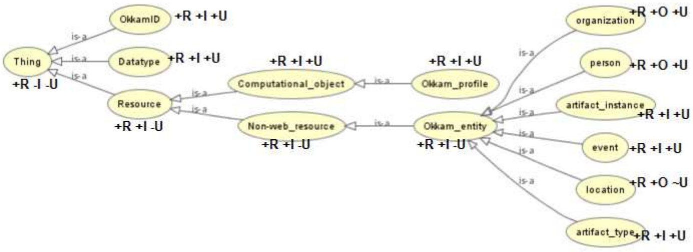


Figure 5.5: Identification Ontology backbone Taxonomy obtained by the Analysis of the Okkam Conceptual Model

ogy. The analysis highlighted some modeling error in the definition of the backbone taxonomy. Some of the concepts treated in the Okkam Conceptual Model, despite intuitively simple, are still under philosophical analysis and non shared agreement exists about their metaphysical nature. In particular big effort was spent in the search for suitable solution to represent events. Despite no shared solution exists, we believe that some important result can be reached considering a more pragmatic approach to the definition of identity and unity property of complex entities as events. It is important to notice that the goal of the project was not to define or defend any specific philosophical position about metaphysics of certain properties. OntoClean showed to be a very useful methodology to analyze the Okkam Conceptual Model, and some of the proposals defined in section 5.2.2 will be used in possible evolution of the model suitable to be adopted as backbone taxonomy for the Identification Ontology supporting the knowledge based solution proposed in the work.

The exercise performed along the analysis of the Okkam Conceptual Model guided the process of dissecting concepts and properties that intuitively appear easy to handle, but that present many subtleties when analyzed in detail with respect to clear identity criteria. The main result of this process with respect to this work is the a further confirmation that the OCM taxonomy is an optimal starting point to define the model the knowledge base supporting the proposed solution. A view of the taxonomy labeled according OntoClean methodology is presented in Figure 5.5.

The taxonomy depicted in figure 5.5 is used as core component of the Identification Ontology underlying the knowledge-based solution for open entity matching. In particular, the part that is more interesting for goal of this work is the part of the taxonomy specifying the Okkam Entity subclasses. However, as a result of the analysis produce in section 5.2.1, we learned that some concepts are particularly slippery with respect to the individualization clear identity criteria. For example, events and artifacts type and instances appear hard to model generally in terms of what type of knowledge is necessary to discern them besides their name. The pragmatic philosophical approach to identity definition used as tool in the application of the OntoClean methodology can hardly support the definition of an explicit knowledge-based solution without relying on deep cognitive studies. An attempt in this direction was proposed in [6], but given the time available and the scope of the thesis, we choose to not investigate deeply these concepts, giving priority to the concepts easier to interpret and manage from an epistemic perspective. For this reason, we choose to explore in detail the definition of a knowledge-based solution for entities of type Person, Location (interpreted as geospatial/political features) and Organization.

5.4 Meta-properties for Identification

As mentioned in the introduction of the chapter, we intend to rely on a set of meta-properties to explicitly highlight special properties of features when involved in the entity matching processes. With this respect, we propose to rely on a top-down ontological analysis to extend the identification ontology attempting to compensate, or integrate, OWL 2 with meta-properties that may be useful to declare properties that have special roles in the matching process. The idea is to apply concepts and principles that are proper of formal ontological analysis (e.g. [70]) to support the definition of matching, or non-matching, constraints. It is important to keep in mind that in formal ontology identity cannot be defined in general as sufficient condition, and what can be defined are actually some sort of information constraints providing necessary condition for identity [69].

As previously mentioned, in the definition of identity criteria OntoClean [70] proposed to consider the distinction between *Synchronic* versus *Diachronic* identity criteria. Synchronic identity criteria allow to establish identity between entities at the same instance of time. Formally, p is synchronic if $\exists t \forall x, y(t > t_0 \rightarrow (p(x, y, t_0) \rightarrow p(x, y, t)))$ with t_0 denoting the first time y was assigned to x as the value of the property p. Diachronic identity criteria imply the notion of persistence of the identity criteria through time. Formally, p is a diachronic if $\forall t, x, y(t > t_0) \rightarrow (p(x, y, t_0) \rightarrow p(x, y, t)))$. Namely, they do not change with time, and thereby represent essential properties of the identified entity. In OntoClean, the individuation of identity criteria is essential to annotate concepts in a taxonomy, and thus evaluate its soundness according to this methodology. However this distinction may become useful also in the context of entity matching problem.

In fact, the solution of the entity matching problem in an open and wide context such as the Web must Diachronic identity criteria have to be preferred with respect to the

Synchronic ones. Indeed, the Web, as global information space, has to be interpreted as intrinsically asynchronous with respect to the real world entities mentioned and mentioned and described in it. Thereby, assume synchronicity of properties for entity matching solution may lead to problems.

As analyzed in chapter 2, the solution of entity matching problem in the context of the web is also prone to the problem of over-specification. Namely, descriptions could contain attributes that can be interpreted only within a specific context, as for example a *student id number*. These properties are very likely to be inverse functional in the context of a university information system. However, there is no guarantee that two systems independently do not assign the same id to different students in different universities. Thereby, there is also a further dimension to be considered con considering identifier: the scope of a property. In particular, we distinguish between local scope and global scope of a property with respect to matching purpose. Intuitively, building a reliable solution for entity matching in a global and open context should preferably rely on properties defined with global scope and treat carefully attributes with local scope only.

With these premises, in the following we define a list of meta-properties considered useful for the solution of the entity matching problem:

- Functional (F): p is functional if for every individual x there can be at most one individual y such that $p(x, y)$[12]. More formally, p is a functional binary predicate: $\forall x, y, z(p(x, y) \wedge p(x, z) \rightarrow y = z$. The official semantic of the functional meta-property is a shortcut for properties with a max cardinality constraints 1 on the range of the property. It is important to underline that the semantic of functional properties is conceived without considering time changing. For example, it sounds correct to say that every person has just one residence address. In fact, any person can be officially be resident only in one place and thus the property can be correctly classifies as functional. However, a person can change residence several time in a life time, and thus considering time dimension, a person can have more than residence address even thou not at the same time.

- Inverse Functional (IF): p is inverse-functional if for every individual y there is at most one individual x such that $p(x, y)$ [13]. More formally, p is inverse-functional predicate: $\forall x, y, z(p(x, y) \wedge p(z, y) \rightarrow z = x$. This type of meta-property is proper of identifiers assigned to entity in some context, and within that context only one entity can be associated to it. Global inverse functional properties are practically impossible as identifiers are usually assigned with a bounded context. However, if

[12] http://www.w3.org/TR/owl2-syntax/#Functional_Object_Properties

[13] http://www.w3.org/TR/owl2-syntax/#Inverse-Functional_Object_Properties

we consider the Web as a global space then email address URIs can be considered as global inverse functional properties. Also in this context, inverse functionality has to be considered in a diachronic manner. Non-diachronic inverse functional properties can cause big troubles when solving entity matching in a global space.

Here we define a set of meta-properties for identification properties, taking into consideration time and scope dimensions. In particular we define:

- Functional Diachronic (FD): p is functional diachronic if and only if it is both functional and diachronic as identity criteria. Namely the range of this property must have cardinality maximum 1, and its value does not change in time for any reason. An example of properties that are functional diachronic for a person are the date of birth and the birthplace.
- Inverse Functional Diachronic (IFD): p is inverse-functional diachronic if it is both inverse-functional and diachronic as identity criteria. Namely, given a property p and a value y, the domain of p must have cardinality maximum 1, and furthermore the value does not change in time.

In the following sections we will describe the process of extending backbone taxonomy produced starting from the OCM with the set of features considered relevant for the identification process.

5.5 Features for Open Entity Matching

In this section we present briefly the methodology adopted to select the set of features associated with each of the three entity types for which we intend to build a knowledge-based entity matching solution. It is important to remember that the features defined for the selected types are parameters of the matching process, and thus in principle any set of features could be used. Intuitively, the broader the set of features allow the broader the set of sources that can be matched precisely. However, a large set of features requires also a large and heterogeneous set of labeled samples presenting these features to learn their relevance in the matching process. Furthermore, the larger the set of attributes the larger and cumbersome is the maintenance of the contextual mappings. Hence, in the short term it is important to rely on an iterative, incremental approach aimed keeping the effort sustainable. Whereas, in a long term perspective, we have to outline a process suitable for a community effort.

In the following sections, we present a set of features obtained relying on different sources and methodologies. Their combination will produce the set of features associated with each of the types considered. The first set of considered features comes from the results of a cognitive science experiment aimed at eliciting through experiments with people what are the most discriminative attributes for several entity types [6, 7, 4]. Essentially, through a set of targeted experiments, the authors of the work elicited what are the attribute types people would use to search for specific type of entities. A statistical analysis of the results based on concepts of *sharedness* and *distinctiveness*, produced a measure of the semantic relevance of the attributes used. The attribute proposed were then ranked based on this estimation, and attributes below a certain threshold were not considered [6]. The results of this work are also used in [139] to weight the relevance of attributes to compute entity matching similarity. This first set of features is the result of methodologically sound cognitive science experiments. However, these experiments were specially focused on eliciting features relevant for searching, and thus identifying entities in the context of the Web. Hence, the set of properties defined is surely relevant, but unlikely to be complete with respect to the need of representing information that could be useful to solve the entity matching problem.

In order to extend this set of properties we surveyed for existing vocabularies and extracted the features the most generic features associated to the considered entity types. Given the large varieties of existing vocabularies presenting similar properties for the the same type of entity, we propose to adopt an approach aiming at clustering shared (similar) properties towards general properties that could encompass all their instantiations. In fact, in generalizing these properties, we considered *sub-property* as a relation supporting clustering of existing properties. Consider for example the properties that provide information about a person belonging to some group, or organization. For a politician, a property indicating the political party of reference could be *dbpedia:party*, whereas for a musician, the band could be indicated with *dbpedia:musicalBand*, or again a noble person could be casted as belonging to a family through the *dbpedia:dynasty* property. It is clear that representing all possible variants of this type of property would lead quickly to an unmanageable vocabulary. Thereby, we decided to cluster these properties to the more generic property: *member_of*. Notice that in principle, this minimalistic approach in the management of the vocabulary has also the important advantage of reducing sparsity in the comparison, but may also introduce some noise as by generalizing we loose some of the specificity embedded in the property definition. Experimental evaluation will show whether the positive effects of generalization compensate the lost of specificity.

The Linked Open Vocabularies[14](LOV) listed 319 vocabulary spaces, providing classifications for and properties for many different types of entities. LOV is a very useful

[14] http://lov.okfn.org/dataset/lov/

entry point for gathering types and properties related to them, as it provides a formal SPARQL interface. Given the large space of vocabularies available only in the Linked Data initiative, we limited our analysis to the most popular vocabularies (e.g. FOAF, Schema.org, DOLCE Lite, etc). The complete exploration of the vocabularies space will be pursued in future work. Noticeable, there are ontologies such as the one of DB-Pedia, Freebase and Yago that are not mentioned in LOV, but we decided to include them anyway.

5.5.1 Features for Entity Type Person

The identification ontology includes the following properties (in alphabetic order) for the entity type Person:

- *affiliation.* This feature describes generically the affiliation of a person with respect to some organization that can be precisely individualized. This feature is generally functional when conceived synchronically, however it often changes in time. This fact does not make this property suitable to take matching decisions alone. However, when considered in an aggregated form, it may be useful to take matching decisions. This property clusters generically properties such as *team*, *company*, *employer*, *workplace*, *military unit*, *party*, etc. The affiliation of a person is not functional, nor inverse-functional.

- *author.* This feature describes the intellectual artifacts (books, plays, software, etc) that are ascribable to a person. The authorship of an intellectual artifact is not functional, nor inverse functional as many authors can contribute to an opera. However, very unlikely there are homonyms co-authoring an opera, and thus this property can be useful to establish matching decisions. Examples of properties clustered around this property are: *writer*, *lyrics written*, *plays_composed*, *artworks*, *novels*, *contributing author*, etc. Similar reasoning was done on many works working on matching authors in bibliographic datasets, e.g. [16, 17].

- *award.* This feature describes awards achieved by a person including sport titles or medals of honors. The property of receiving an award is neither functional, nor inverse functional if considered in an asynchronous context as many person can receive the same award at the same of different time. However, for most of the cases, very unlikely there are homonyms receiving the same award at the same time, and thus this property can be useful to establish matching decisions. Examples of properties clustered around this property are: *prizes*, *award*, *honored for*, *Olympic medals*, *best player of the game* etc.

- *birth date.* This feature describes the date of birth of a person (e.g. 12 Feb 1982). This property is functional, included among the properties described in [6], and generally used also in standard identification documents. More importantly, the birth date of a person is also functional and diachronic. In fact, the known birth date of a person can never change in time.

- *birth month.* This property describes the month of the year part of the date of birth of a person (e.g. February). This property, as part of the birth date, is functional and diachronic. It is included mostly to ease problems related to structural heterogeneity in the representation of date of birth values. For example, some sources would not present the date as single attribute, but would rather present the three parts of a date of birth separated in the day, month and year. As one of the goal of the ontology is to provide a mean for semantic harmonization, loosing information about the date of birth out of granularity issues would affect the quality of matching.

- *birthplace.* This feature describes the place or location where a person was born. This property is functional, included among the the properties described in [6], and generally used also in standard identification documents. As for the birth date, the birthplace is a functional and diachronic property. However, identifying precisely a place may be more complicated than a date, as location names are generally ambiguous.

- *birth year.* This feature describes the year as part of the date of birth of a person. As for *birth month*, this property is included to ease issues related to structural heterogeneity. Furthermore, for many persons (e.g. historical persons) the exact date of birth may not be available or known, but the year of birth is more likely to be known. As for birth date and birth month, also the birth year is functional and diachronic as it can never change in time.

- *country of residence.* This feature describes the name of the country where a person is resident. This property is functional, even thou it can change in time. This property is usually included in standard identification documents, and usually included as part of the street address of a person. We choose to represent it also as part of this composite attribute with the goal of ease the issues related to structural heterogeneity. Differently from city names, country names are quite reliable for matching as, at best of our knowledge, do not exists different countries with the same name.

- *city of residence.* This feature describes the name of the city where a person is resident. This property is functional, even thou it can change in time. This property is usually included in standard identification documents, and usually included as part of the street address of a person. We choose to represent it also as part of this composite attribute with the goal of ease the issues related to structural heterogeneity. As for birthplace, the identification of a city is complicated due to the inherent ambiguity of city names when interpreted in a global space.

- *date of death.* This feature describes the date of death of a person. This property is functional and diachronic, and usually very useful to take matching decision about historical person when available. As for birth date, the date of date can never change in time.

- *day of birth.* This feature describes the day of the month as part of the date of birth of a person (e.g. 12). This property is functional, and it is included mostly to ease problems related to structural heterogeneity in the representation of date of birth values. For example, some sources would not present the date as single attribute, but would rather present the three parts of a date of birth separated in the day, month and year. As one of the goal of the ontology is to provide a mean for semantic harmonization, loosing information about the date of birth out of granularity issues would affect the quality of matching. As for any other part of a birth date, the property is also diachronic.

- *deathplace.* This feature describes the place of death of a person. This property is functional and diachronic, potentially useful to take matching decisions about historical person when available. Furthermore, the property is also diachronic, as the place of death can in principle never change in time. However, given the ambiguity related to the identification of a place, we have to treat carefully the matching of this type of attributes.

- *description.* This feature aims at being a placeholder for all the attributes providing generic, textual descriptions about a person. This feature clusters properties such as *short description*, *abstract*, *notes*, *biography summary*, etc. Any property valued with textual description about a person should be mapped towards this feature. A description usually contains information in natural language that can help human being in taking matching decisions. However, natural language is known to be very ambiguous and hard to interpret. For this reason, we have to deal carefully with this type of feature.

- *domain tag.* This feature is aimed at representing an alternative, compact type of description. In fact, domain tag feature aims at collecting all those attributes that may be useful to describe a person in terms of keywords. Data sources in the web can use this type of attribute to contextually disambiguate records about homonyms without the need of providing any specific semantic. The *domain tag* feature clusters properties such as all the generic *tag* and *domain*, and also *sport played*, *music genre played*, etc.

- *email address.* This feature aims at representing the email address of a person. An email address is often used as identifier in web system, as it is generally considered to be inverse-functional. Each email address, as defined according to URI standards, can be reliably considered unique in a global scope. However, mail boxes can be re-assigned in time and thus in principle they could not be considered inverse-functional and diachronic. However, for how formally correct would be this conclusion, we believe that in general email addresses are still very reliable as web identifiers, even thou we are aware that this could lead to possible erroneous match decision. For the moment, we decide to introduce a simplification assumption, neglecting the lifecycle of email address and other types of URIs with respect to their assignment to different person in time. A similar choice was made while modeling the FOAF ontology, assuming inverse-functionality of mail box addresses[15].

- *email address hashcode.* This feature aims at representing the result of the encryption of email addresses using hashing algorithms. This property is inspired by the FOAF mbox_sha1sum[16] property of an Agent. "*The sha1sum of the URI of an Internet mailbox associated with exactly one owner, the first owner of the mailbox*". In [13], Berners Lee cites this technique as an approach to link safely different FOAF profiles without disclosing private information. Also in this case, we can assume that an email address is inverse-functional, but as for email address, we cannot conclude that it is also diachronic. Nevertheless, as for email address, we decided that in this context we can neglect this aspect, and introducing a simplification with respect to the complexity of the real world, consider also email address hashcode checksum as inverse functional and diachronic.

- *end date.* This feature aims at capturing a date that represent an end of an activity or mandate. For example, the end of an elective mandate, the end of a period of affiliation with some organization, or the end of activity in some sector (e.g. sport). This information alone does not help in capturing generic knowledge

[15] `http://xmlns.com/foaf/spec/#term_mbox`: A personal mailbox, ie. an Internet mailbox associated with exactly one owner, the first owner of this mailbox. This is a 'static inverse functional property', in that there is (across time and change) at most one individual that ever has any particular value for foaf:mbox.

[16] `http://xmlns.com/foaf/spec/#term_mbox_sha1sum`

related to the context of interpretation. However, it could contribute to some sort of time related inference/reasoning supporting matching decisions. E.g. if the end date of an activity is posterior to *date of death* or anterior with the *date of birth*, then it is possible to produce a negative decision. However, we do not explore this type of constraints in this context, and we limit the analysis to its adoption as an attribute for matching *per se*.

- *eyes color*. This feature describes the colors of the eyes of a person. It is included among the one obtained through experiments with people described in [6]. In fact, this property is functional and generally never changes in the life of a person, and can be considered functional and diachronic.

- *fax number*. This feature describes the personal fax number of a person. This property is likely to be found in data sources describing public officers, together with phone number. There exist standard for the representation of these types of attributes as URIs[17]. This property intuitively is inverse-functional when refers to the personal fax machine of a person. However, this type of machine sometimes can refer to different person at different times, or collectively be used by a group of people (e.g. an office room). Therefore, fax numbers have to be treated carefully. However, knowing if a fax number is personal or of a group is an epistemic problem we can neglect in this context. Hence, aware of this simplification, we consider fax number as inverse-functional and diachronic property.

- *first name*. This feature indicates the first name (or given name) of a person. Any person has at most a first name, which has to be intended as the part of the name of the person without the surname (or family name). A first name is intuitively functional and extremely rarely changes in time. Thereby, it can be considered functional and diachronic. Also in this case, we are aware of the fact that by law a person can change official first name, but we decide to simplify the interpretation of this type of attribute by neglecting this possibility.

- *gender*. This feature describes the gender of a person (e.g. male and female). Any person can belong at most to one gender at time, thus gender is clearly functional. We are aware there exists exceptions, but in this context we assume that the gender of a person is stable in time and neglect to consider exceptions. Therefore, also in this case, we assume that gender is functional and diachronic.

- *height*. This feature describes the height of a person. This property is functional, but it cannot be considered diachronic in general. It is included among the proper-

[17] `http://www.ietf.org/rfc/rfc2806.txt` - URLs for Telephone Calls

ties described in [6], and generally used also in standard identification documents. Nevertheless, the fact different measuring system exists (e.g. metric VS imperial) does not allow us to treat the value of this property easily as functional.

- *involved in.* This feature describes the generic relation of a person with some product or event. This relation is aimed at clustering relations such as *movie (or play, or tv episodes) producer*, *movie (or play) director*, *music producer*, etc. In a sense, the person is involved in the realization of something, but does not participate directly (or practically) in its realization. This property is not functional, nor inverse-functional. Nevertheless, it can be useful in taking matching decisions in combination with other attributes. For example, it is very hard that two homonym produce the same movie, or direct the same play.

- *last name.* This feature describes the family name of a person. We make no assumptions about the nature of the size or number of elements represented by this feature. For example, Portuguese names usually include both father and mother last names. In this context we consider as last names the complement of *first name* attribute in composing the full name of a person. The last name of a person is generally a functional property. However, differently from first name, social conventions related to marriage can easily lead to a change of surname. In this case we do not feel like introducing any simplification, and simply accept the fact that surname cannot be reliably used as an attribute functional and diachronic.

- *member of.* This feature describes the belonging of a person with respect to some group or organization. The *member of* relation is a weaker, less formal/subordinate interpretation of the *affiliation* relation. This feature clusters properties like, *movement*, *school tradition*, *religion*, etc. to which a person can be associated to. The property is neither functional nor inverse-functional, but can help in taking matching decision as a further element of similarity or difference.

- *middle name.* This feature describes possible middle name of a person. This type of feature is included to ease issues related to structural heterogeneity. In fact, some sources could present the parts of the name of a person into separate parts without representing them explicitly as one. Representing the part of the names would allow to reconstruct the whole name of a person based on simple syntactic rules. This property is functional and diachronic as first name.

- *name.* This feature describes the name of a person. The name has to be included as the whole name of a person, including first, middle and last name(s). Given the non strictly diachronic nature of the surname attribute, we cannot assume that *name* is a functional and diachronic attribute.

- *nationality.* This feature describes the nationality of a person. This attribute was among the one obtained through experiments with people described in [6]. The nationality of a person is quite a vague concept, as it often involves the concept of citizenship. One can have multiple citizenships, and change it in a life time when migrating from a country to another. For this reason, the property cannot be considered neither functional nor inverse-functional.

- *nickname.* This feature describes the nickname of a person. A nickname is any name defined and used in a private unofficial context (e.g. skype.id), the name in art (or alias) of some artist (e.g. Alice Cooper born Vincent Damon Furnier), or the royal name (e.g. Queen Elisabeth 2nd). A nickname cannot be considered functional, nor inverse-functional out of its context. However, together with other attributes, it can contribute in taking matching decision as a nickname in the case of famous person, can be better known than a first name.

- *occupation.* This feature describes the main occupation of a person. It is clear that this type of information is useful to take matching decisions if occupations are incompatible (doctor vs football player). However, this type of inference is very much context dependent and very in depth formalization of the different types of occupation is required to automatize it. In a life time, a person changes occupations several times, and thus it cannot be considered functional and diachronic. This attribute was among the one obtained through experiments with people described in [6], and it clusters the attributes ranging from *basketball roster position*, to *instrument played in a band* and *government position held.* More generically, any property denoting the *role* of a person.

- *participant in.* This feature describes generically the participation of a person to some event. This property is neither functional nor strictly inverse-functional. In fact, depending on the type of event, being part of it does not guarantee the possibility of identifying a person. However, if we have description presenting the same name and participating in the same concert or play, we are very likely to assume that the two descriptions refer to the same person. The participation to an event is necessarily diachronic as identity criteria, despite the identification of the event is quite slippery besides traditional recognized ones (e.g. the birth of a person). This feature clusters properties such as *movie appearance*, *sport competition*, *battles*, *musician tours*, *sport season or matches*, *election campaign*, *legislative session*, etc.

- *phone number*. This feature describes the phone number of a person. Similarly to *fax number*, this property is likely to be found in data sources describing public officers. There exist standard for the representation of these types of attributes as URIs[18]. This property intuitively is inverse-functional when refers to the personal phone number of a person (e.g. mobile phone). However, often phones refer to different person at different times, or collectively be used by a group of people (e.g. an office room, family phone, etc). Nevertheless, we assume that a phone number is person neglecting for the collective phone number interpretation and the fact that a phone number can, in principle, be assigned to different person in time.

- *picture URL*. This feature describes the URL of the picture of a person. This property is inverse-functional if the picture depicts only one person. In fact, in principle a picture could depict more than a person. However, intuitively it is very unlikely that two person with the same name use the same picture URL in which they appear. Also in this case, we choose to introduce a simplification assuming that any picture refers specifically to the person, and thus it can be considered inverse functional and diachronic.

- *postal code*. This feature describes the postal code of a person. This property is functional, but can change with time. It is represented as single feature to ease the problems related to structural heterogeneity related to the representation of street addresses.

- *start date*. This feature aims at capturing a date that represent an start of an activity or mandate. For example, the start of an elective mandate, the start of a period of affiliation to some organization, and so forth. As for *end date*, this information per se does not help in capturing generic knowledge related to the context of interpretation. However, it could contribute to perform time related inference/reasoning supporting matching decisions. E.g. if the start date of an activity is posterior to *date of death* or anterior with the *date of birth*, then it is possible to produce a negative decision. However, we do not explore this type of constraints in this context.

- *street address*. This feature describes a street address referred to a person. Similarly to phone and fax number, this type of information is likely to be found in data sources describing public officers, or health care providers. A street address

[18] `http://www.ietf.org/rfc/rfc2806.txt` - URLs for Telephone Calls

is not functional, nor strictly inverse functional, as a person can change street address when moving, and many person can live at the same address. Thereby, it cannot be considered neither functional nor inverse-functional and diachronic. However, as for other attributes, it is very unlikely the two homonym live at the same address, and thus, street address can help in taking matching decisions in combination with other attributes.

- *public institutional id.* This feature describes the possible identifiers assigned to a person by a public institution. Despite these types of attribute are not granted to be globally unique as no common convention is shared, these are considered to be inverse-functional and diachronic neglecting the very unlikely occasion where two strings are created by different institutions to be equal and refer to the different real world person. This feature clusters *social security number*, *tax code*, *driving license number*, etc.

- *title.* This feature describes the title of a person. The feature encompasses *honorific prefixes*, *noble titles*, and *seniority prefixes.* This type of property is not functional, nor inverse-functional. However, together with other types of attributes, it can help in taking accurate matching decisions.

- *website.* This feature describes the URL of the web page about a person. This feature is inverse-functional if the URL refers to a personal web page. However, domain name are not granted to refer to the same web page in time. In fact, it is possible that a domain name once held by a person, is then used by another one when the former owner of the domain name let the registration subscription expire or simply sells it. Despite this, matching personal web pages are very likely to refer to the same real world person. This feature describes also the URL of web pages of a person related to accounts on social media applications or platforms (e.g. twitter, facebook, linkedIn, etc). This type of information is considered inverse-functional as the application context guarantees the inverse-functionality of the URL. Even thou personal web sites cannot be considered strictly diachronic, we decide to neglect it and assume that websites are inverse-functional and diachronic.

5.5.2 Features for Entity Type Location

The identification ontology includes the following properties (in alphabetic order) for the entity type Location:

- *area.* This feature describes the area of a location according to some measure system (meter, foot, etc.). The area of a location is intuitively a functional property.

However, as discussed in section 5.2.1, locations area is not an identity criteria proper of locations per se but are the result of some definition process. Namely, someone or some authority defines the borders of locations that then allow to estimate an area. This implies that in time, different definitions can change the value of the attribute, making it not reliable for taking matching decisions. However, to a certain extent, this attribute can be useful to take matching decision when considered together with other attributes. Therefore, we do not consider the area of a location as functional diachronic property.

- *city.* This feature describes the name of a city of a location. This features aims at representing the location containing the described location at a level of granularity of a city, or town. The city is often used as part of the postal address to refer precisely to a location. The city of a location is a functional property, as any location can be contained at most in one city. Therefore, we assume that the city of a location is a functional and diachronic attribute.

- *contains.* This feature describes possible location or geographical features contained, or partially contained, by the described location. For example, a location city may contain several districts or neighborhood. A region contains cities, etc. The locations contained by a location may be useful to distinguish the location itself from the ones contained when the information are available. The contains property is neither functional nor inverse-functional as a location can contain many locations, and the same location can also be contained by other locations.

- *coordinate geometry.* This feature describes the geographical coordinate of a location interpreted as geometrical points rather than points in a coordinates system. In fact, coordinates system usually present *latitude* and *longitude* in this precise order, whereas when we want to represent them on a Cartesian plane, we represent first longitude as it represent the value changing on the horizontal dimension and then latitude that represent the value on the vertical dimension. The coordinate geometry value represent thus the coordinates of a location represented according to the Cartesian system. Coordinate geometry property are functional and quasi-inverse-functional. Namely, many locations could share the same coordinates. For example a building may contain several apartments or shops which would share bi-dimensional coordinates. However, if the coordinates match and the name of location matches, we can easily conclude it is the same location.

- *country.* This feature describes the name of the country of a location. This features aims at representing the location containing the described location at a level of granularity of a country. As for the *city*, country information is often used as part of the postal address to refer precisely to a location. The country of a location is a functional property, as any location can be contained at most in one country.

- *description.* As for other entity types, this feature generically denotes textual description about a location.
- *domain tag.* This feature is aimed to represent an alternative, compact type of description. In fact, domain tag feature aims at collecting all those attributes that may be useful to describe a person in terms of keywords.
- *elevation.* This feature describes the elevation of a location with respect of some measure system (e.g. meter, foot, etc.). The elevation of a location is intuitively a functional property, and despite can slightly change in time, it can be considered diachronic. However, when referring to a location, it is important to consider how the elevation measure is estimated. In fact, the elevation of a point can be considered clearly functional and diachronic, but we cannot take the same conclusion when we consider an area.
- *first level administrative parent.* This feature describes the first level of subdivision of the country containing the location described. The first level of subdivision changes according to the country. For examples, Italian's first level subdivision levels are regions, whereas for federal states like Germany, USA or Brazil the first level of subdivision are the states, etc. This property is functional, as any location can be contained at most by one first level administrative subdivision, if any.
- *second level administrative parent.* This feature describes the second level of subdivision of the country containing the location described. The second level of subdivision changes according to the country. For examples, Italian's second level subdivision levels are provinces, whereas for federal states like USA second level of subdivision are the counties, for Germany regions, etc. This property is functional, as any location can be contained at most by one second level administrative parent, if any.
- *third level administrative parent.* This feature describes the third level of subdivision of the country containing the location described. The third level of subdivision changes according to the country. For examples, Italian's third level subdivision levels are municipalities, whereas for federal states like Germany third level of subdivision are district, etc. This property is functional, as any location can be contained at most by one third level administrative parent, if any.

- *forth level administrative parent.* This feature describes the forth and most fine grained level of subdivision of the country containing the location described. The third level of subdivision changes according to the country. For examples, Italian's forth level subdivision levels are fractions or *circoscrizioni*, whereas for federal states like Germany third level of subdivision are municipalities, etc. This property is functional, as any location can be contained at most by a forth level administrative parent, if any.

- *geocoordinate.* This feature describes the combination of latitude and longitude of a location to indicate a point in a coordinate system. This property is functional, and as for *coordinate geometry*, it can be inverse-functional when combined with the location name attribute.

- *is contained by.* This property denotes the name of the location containing the described location. This property is a sort of bulk container that encompasses all the administrative subdivisions listed above, and includes also levels of subdivision not considered (e.g. group of island, neighborhood, etc). As the level of the location containing the describe location is not explicitly known, it is not possible to consider this property as functional. Possible secondary sources can be exploited to categorize containers with respect to their corresponding administrative level, if any, as proposed in [109]. This property cannot be considered functional as many location could be represented as containing another one. Furthermore, the property cannot be considered inverse-functional as each of the containing location can contain more than one location.

- *latitude.* This property describes the geographic coordinate latitude of a location. Latitude specifies the north-south position of a point on the Earth with respect to the equator. The numerical value of the latitude can change according to the coordinate system. Some systems are based on the surface of the earth (e.g. average level of the sea), on which is approximated a geometrical shape (e.g. ellipsoid), then used to compute an angle between the radius from the center of the geometrical shape to the point and the radius of the point on the equator. There are many approximation of the surface of the Earth, and of its shape. Several standards have been defined and applied to different contexts (e.g. military standard). In most of the cases, knowing the reference system, it is possible to transform the coordinates from one system to another before comparing them. However, this type of contextual information will be seldom available on the web. The latitude of a location is functional and diachronic, as it cannot change in time.

- *latitude degree.* This feature describes the degree of a latitude coordinate (e.g. 10°). Some data sources may present latitude data at a level of granularity distinguishing each of the parts of the coordinate. Representing the features at this level of granularity would allow to ease the problems related to structural heterogeneity and improve comparison quality. The latitude degree is a functional and diachronic property.

- *latitude direction.* This feature describes the direction of the latitude coordinate with respect to the equator (e.g. north or south). Some data sources may present latitude data at a level of granularity distinguishing each of the parts of the coordinate. Representing the features at this level of granularity would allow to ease the problems related to structural heterogeneity and improve comparison quality. The latitude direction is a functional and diachronic property.

- *latitude minute.* This feature describes the minute of the latitude coordinate (e.g. $10'$). Some data sources may present latitude data at a level of granularity distinguishing each of the parts of the coordinate. Representing the features at this level of granularity would allow to ease the problems related to structural heterogeneity and improve comparison quality. The latitude minute is a functional and diachronic property.

- *latitude second.* This feature describes the minute of the latitude coordinate (e.g. $10''$). Some data sources may present latitude data at a level of granularity distinguishing each of the parts of the coordinate. Representing the features at this level of granularity would allow to ease the problems related to structural heterogeneity and improve comparison quality. The latitude second is a functional and diachronic property.

- *location name.* This feature describes the name of the location (e.g. Trento). This attribute clusters all possible attributes presenting the name of a location, including ISO 3166 country codes[19], alternative names, etc.

- *location type.* This feature describes generically the type of a location. This feature clusters properties like *category*, *type*, *feature class*, representing information related to the possible categorization of the location.

- *longitude* This property describes the geographic coordinate longitude of a location. Longitude specifies the east-west position of a point on the Earth with respect to the Greenwich meridian. The numerical value of the longitude can

[19] http://www.iso.org/iso/country_codes.htm

change according to the coordinate system. As for the latitude, some systems are based on the surface of the earth (e.g. average level of the sea), on which is approximated a geometrical shape (e.g. ellipsoid), then used to compute an angle between the radius from the center of the geometrical shape to the point and the radius of the point on the meridian. There are many approximation of the surface of the Earth, and of its shape. Several standards have been defined and applied to different contexts (e.g. military standard). In most of the cases, knowing the reference system, it is possible to transform the coordinates from one system to another before comparing them. However, this type of contextual information will be seldom available on the web. The longitude of a location is a functional and diachronic property.

- *longitude degree*. This feature describes the degree of a longitude coordinate (e.g. 10°). Some data sources may present longitude data at a level of granularity distinguishing each of the parts of the coordinate. Representing the features at this level of granularity would allow to ease the problems related to structural heterogeneity and improve comparison quality. The longitude degree of a location is a functional and diachronic property.

- *longitude direction*. This feature describes the direction of the longitude coordinate with respect to the Greenwich meridian (e.g. east or west). Some data sources may present longitude data at a level of granularity distinguishing each of the parts of the coordinate. Representing the features at this level of granularity would allow to ease the problems related to structural heterogeneity and improve comparison quality. The longitude direction of a location is a functional and diachronic property.

- *longitude minute*. This feature describes the minute of the longitude coordinate (e.g. $10'$). Some data sources may present longitude data at a level of granularity distinguishing each of the parts of the coordinate. Representing the features at this level of granularity would allow to ease the problems related to structural heterogeneity and improve comparison quality. The longitude minute of a location is a functional and diachronic property.

- *longitude second*. This feature describes the minute of the longitude coordinate (e.g. $10''$). Some data sources may present longitude data at a level of granularity distinguishing each of the parts of the coordinate. Representing the features at this level of granularity would allow to ease the problems related to structural heterogeneity and improve comparison quality. The longitude second of a location is a functional and diachronic property.

- *picture URL.* This feature describes the URL of a picture depicting a location. As URL, value of this attribute is granted to be globally unique in the web space. As for person, there are many pictures that can be depicted in a picture. However, in this case we assume that picture refers specifically to the described locations and thus it can be considered as inverse functional and diachronic.

- *postal code.* This feature describes the postal code of a location. This feature, in combination with *city*, *country* and *street address* allows to identify quite precisely a location on the earth. Unfortunately, postal codes may change in time according to possible reformation of the postal system. However, representing the features at this level of granularity would allow to ease the problems related to structural heterogeneity and improve comparison quality. Furthermore, a location may contain more than one postal code, and many locations can have the same postal code. For this reason, we do not consider postal code neither as functional nor as inverse-functional.

- *street address.* This feature describes the street address property of a location. A complete street address for a location can be considered inverse-functional. In fact, a street address identifies uniquely a geographical feature (e.g. building, or property), as well as the possible sub-locations that can be contained as part of this location. However, street addresses follow different standard in different parts of the world, and their matching is known to be a complicated problem from a syntactic point of view. Exploiting secondary resources as proposed in [108] could be a viable option to ease this problem.

- *timezone.* This feature denotes the time zone region containing the location described. The time zone of a location cannot be considered functional in general, as there exists nations including different time zones (e.g. USA, Russia, China, etc.). However, these large countries are also very likely to contain many homonym city names, and thus when properties about parent subdivision cannot be interpreted precisely, difference in time zone can be used to support matching decisions.

- *website* This feature describes the URL of the web page about a location. This feature is inverse-functional and diachronic if the URL refers to a web page of the administrative organs of the location or the about the location itself.

5.5.3 Features for Entity Type Organization

The identification ontology includes the following properties (in alphabetic order) for the entity type Organization:

- *activity sector*. This feature describes generically the sector or field in which an organization is active. Examples of sectors are industrial sectors, education, gastronomy, tourism, etc. This feature clusters properties such as *sport* for a team, *field of study* for an education institute, *ideology* for a party, *musical genre* for a band, *medical specialities* for an hospital, etc. This property is neither functional nor inverse-functional, but it allows people to perform some sort of inference based on how compatible are the activity sectors as part of organizations descriptions.

- *activity start year*. This feature describes the year in which an organization started operating actively. This property is functional and diachronic as the *birthyear* of a person, and it is represented at this level of granularity to ease the problems related to structural heterogeneity in the representation of this type of information (i.e. dates).

- *associated with*. This feature describes a generic association of an organization with other entities (e.g. organization, brands, etc). This feature clusters properties such as *associated acts* for musician, *brand*, *spin-off*, *affiliation*, *partner*, *sister companies*, etc.

- *award*. This feature describes the award or prices won by an organization. The property of receiving an award is neither functional, nor inverse functional as many organization can receive the same award. However, for most of the cases, very unlikely there are homonyms receiving the same award at the same time, and thus this property combined with other can be useful to establish matching decisions. Examples of properties clustered around this property are: *prizes*, *award* for bands, *titles* in sport competition, *best product* etc.

- *city*. This feature describes the name of the city where an organization operates. This property is not functional for companies that operate in different cities, and it can change in time. This property is usually included as part of the street address of an organization. We choose to represent it also as part of this composite attribute with the goal of ease the issues related to structural heterogeneity.

- *color*. This feature describes the colors that are associated with an organization. This property is not common to any organization, but it is proper of sport teams, or schools. The property is neither strictly functional nor inverse-functional but it can help in taking matching decisions when two descriptions present the same football team name such as Inter of Milan in Italy, and Inter of Porto Alegre in Brazil, but the colors are different. Namely, black and blue for the first, and red and white for the second.

- *controlled by.* This feature describes the name of the controller of an organization. This property is not functional, and could be considered inverse-functional without considering time dimension. However, organizations can be acquired and sold continuously, so this property cannot be considered reliably inverse-functional and diachronic. This property clusters properties such as *parent company*, *acquiring organization*, and so on. Despite it cannot be considered inverse-functional and diachronic, it can still be used to take matching decision.

- *controls.* This feature describes the name of organizations controlled by the one described. This property is not functional, and could be considered inverse-functional without considering all the controlled entities and time dimension. However, organizations can be acquired and sold continuously, so this property cannot be considered reliably inverse-functional and diachronic. This property clusters properties such as *child company*, *organization acquired*, *holding*, and so on. Despite it cannot be considered inverse functional, this type of features can be considered useful to take matching decisions.

- *country.* This feature describes the name of the country where a company is located. This property is not functional for international companies, and it can change in time. This property is usually included as part of the street address of a company. We choose to represent it also as part of this composite attribute with the goal of easing the issues related to structural heterogeneity.

- *description.* This feature denotes a descriptive text about an organization. This feature clusters properties such as *abstract*, *description*, *biography*, etc.

- *dissolution date.* This feature describes the date of dissolution date of an organization. This feature is functional and diachronic and useful take matching decision about historical companies, or to distinguish different companies with the same name but operating at different times.

- *domain tag.* This feature, as for person and location, aims at presenting keywords describing an organization.

- *email address.* This feature describes the email address of an organization. Differently from people, this type of information is often available online and can help in taking precise matching decisions as email addresses are inverse-functional and diachronic with respect to the organization.

- *email address hashcode*. This feature aims at representing the result of the encryption of email addresses using hashing algorithms. This property is inspired by the FOAF mbox_sha1sum[20] property of an Agent. "*The sha1sum of the URI of an Internet mailbox associated with exactly one owner, the first owner of the mailbox*". As organizations are usually less concerned about privacy, this type of property is likely to be seldom used. As an email address, also this property is considered inverse functional and diachronic.

- *end date*. This feature aims at capturing a date that represent the end of an activity. For example, the end of an activity in a country, the end of a period of affiliation with some organization, or the end of activity in some sector (e.g. sport). This information alone does not help in capturing generic knowledge related to the context of interpretation. However, it could contribute to some sort of time related inference/reasoning supporting matching decisions. E.g. if the end date of an activity is posterior to *dissolution date* or anterior with the *foundation date*, then it is possible to produce a negative decision. However, we do not explore this type of constraints in this context.

- *fax number*. This feature represents the fax number of an organization. This type of property is inverse-functional and diachronic with respect to an organization. In fact, fax number are often necessary for effective document communications and thus are also often publicly available information.

- *foundation date*. This feature describes the foundation date of an organization. This property is functional and diachronic.

- *founded by*. This feature describes the name of the persons founding the organization. The property is neither functional, nor inverse-functional, but it seems very unlikely that two person with the same name found organizations with the same name. Thereby, in aggregation with other information, this type of information can be very useful to take matching decisions when available.

- *geocoordinate*. This feature describes the geocoordinate of the main building where an organization is operating. This type of information is functional but not inverse-functional, as many organizations can operate in the same building. However, if we consider time dimension, the main building of an organization can

[20] `http://xmlns.com/foaf/spec/#term_mbox_sha1sum`

change and thus its functionality is not reliable. Nevertheless, in combination with other features this property can be very useful to take matching decisions.

- *has foundation place.* This feature describes the location where an organization was founded for the first time. This property is functional and diachronic, and can be considered as the *birthplace* of an organization. However, given the inherent ambiguity of locations names when interpreted in the context of the web, these property has to treated carefully.
- *has key people.* This feature describes the name of a person occupying a key (important) position in an organization. This property is neither functional nor inverse functional, but can be useful to take matching decision in combination with other attributes. The feature clusters properties like *coach*, *CEO*, *editor*, *manager*, *president*, *commander* and any property describing the name of the person in a leading role.
- *has location.* This feature describes generically the name of a location in which a location is located. This feature clusters properties like *region server*, *school district*, *neighborhood*, *contained by*, *location*, etc. This property is not functional nor inverse functional, but it can help in taking matching decision together with other attributes.
- *has members.* This feature describes the names of the person that are known to be member of an organization. This property is neither functional, nor inverse-functional taken singularly. However, considering all members together could be useful to support matching decision. This features clusters properties such as *member of a band*, *employees*, *players*, *roster*, *students*, etc.
- *has parts.* This feature describes the name of an organization that is parts of the described organization. This property is not functional, and in general could be considered inverse-functional if it would possible to identify precisely the part. However, in general organizational parts of companies are not granted to have a unique name (e.g. agencies, department, etc.) per se. Thus is seems risky to consider them inverse-functional. This feature clusters properties like *departments*, *divisions*, *units*, *branches*, *bodies* and so on.
- *involved in.* This feature describes the involvement of an organization with respect to some event. This property is neither functional nor inverse-functional, but in combination with other attributes it could contribute to matching decisions. This feature clusters properties like *sponsored festival*, exhibitions or *conferences*, *featured movies*, *convicted in court*, *involved in public scandal*, and so on.

- *is part of.* This feature describes the name of some organization (or institution) to which the company participate. Namely, every organization that is part of a larger organization, indicates the through this property the name of the organization it is part of. This features clusters properties like *league*, *record label*, *parent institution*, *is a component of*, and so on.

- *latitude.* This feature describes the latitude of the main building where an organization is operating. This type of information is functional but not inverse-functional, as many organizations can operate in the same building. However, if we consider time dimension, the main building of an organization can change and thus its functionality is not reliable. Nevertheless, in combination with other features this property can be very useful to take matching decisions.

- *longitude.* This feature describes the longitude of the main building where an organization is operating. As for the latitude and generally geocoordinates, this type of information is functional but not inverse-functional, as many organizations can operate in the same building. However, if we consider time dimension, the main building of an organization can change and thus its functionality is not reliable. Nevertheless, in combination with other features this property can be very useful to take matching decisions.

- *name.* This feature describe any name of the organization. This property clusters properties like *organization legal name*, *company name*, *operative name*, and so on. Legally, a company can have at most one name. However, it is not uncommon to refer to a company through brand name, or other type of names. Thereby, we cannot assume, when interpreted on the web, that the name of a company is functional and diachronic.

- *nationality* This feature describes the nationality of an organization. This property was included among the one result of the experiment described in [6]. Many companies nowadays operate in multi-national context, thus nationality can hardly be interpreted in this context.

- *offers.* This feature describes the name of anything offered by an organization. This property is aimed at clustering all the properties related to products manufactured, or services provided by the organization. For example, properties such as *products*, *event organized*, *album played*, *drugs*, *rockets*, *computers*, *software programs*, and so on. In principle, if the name of the product is not a trade mark, we cannot assume that a product name is inverse functional. However, it seems very unlikely that two different companies with the same name produce also offer also products with the same name. Hence, this type of attribute can help in taking matching decision together with other attributes.

- *organization type*. This feature describes the legal type of organization (e.g. non profit, private school, public company). This feature is neither functional nor inverse functional, but it can support matching decision in combination with other attributes.

- *participant in*. This feature describes the name of an event to which the organization participated as such. Consider for example concerts or festival for bands, battle for military units, legislative sessions for political parties.

- *phone number*. This feature describes the phone number of an organization. Similarly to *fax number*, this property is likely to be found in publicly data sources about companies. As previously mentioned, there exist standard for the representation of these types of attributes as URIs[21]. This property intuitively is inverse-functional when refers to an organization.

- *picture URL*. This feature describes the URL of a picture depicting an organization or group. A picture can hardly depict an organization, apart for music bands or sport teams. However, pictures could depict organization logos, or brands that could help in identifying the organization. Anyway, we assume that an URL of the picture of an organization is inverse-functional and diachronic, and thus that is explicitly refers to the organization.

- *postal code*. This feature describes the postal code of a company as part of the street address. The postal code of main street address is functional, but can change in time as changes the street address of the organization. Nevertheless, considering the postal code can help in taking matching decisions.

- *previous name*. This feature describes the previous names of an organization. In fact, an organization can change name in time and due to merging, fusion, or simply to renew the brand. This type of information may be useful to take matching decision.

- *public institutional id*. This feature describes the public institutional id assigned by national authorities to an organization. This property is inverse-functional in combination with the country, and clusters properties like *VAT Number*, *tax code*, etc. Inverse-functionality is guaranteed within national borders, but not necessarily outside the border as there is not uniform standard for their definition.

[21] `http://www.ietf.org/rfc/rfc2806.txt` - URLs for Telephone Calls

- *slogan.* This feature describes the slogan of an organization. Slogan, like brands, convening a message help in identifying the an organization. This type of information is not functional, nor inverse-functional, but can be useful in taking matching decisions.

- *start date.* This feature aims at capturing a date that represent an start of an activity for an organization. For example, the start of an activity in a country, the start of a period of affiliation with some organization, or the start of activity in some sector (e.g. sport). This information alone does not help in capturing generic knowledge related to the context of interpretation. However, it could contribute to some sort of time related inference/reasoning supporting matching decisions. E.g. if the start date of an activity is posterior to *dissolution date* or anterior with the *foundation date*, then it is possible to produce a negative decision. However, we do not explore this type of constraints in this context.

- *street address.* This feature describes the street address of the main building where an organization operates. Considering time dimension, the street address is neither functional nor inverse functional. In fact, many companies can operate in the same building, and a company can change street address in time. However, when considered in combination with other attributes it can support matching decision.

- *website.* This feature describes the official domain name of the website of an organization. This property is reasonably inverse-functional, even thou in principle the same domain name could be owned by different organization at different point in time. This feature describes also the social media account (e.g. linkedIn, facebook, twitter) of an organization. Organization may want to be part of social media to keep connection with the members of the organization, or communicate with customers and competitors. Social media URLs are inverse-functional properties as they are defined according to common unique standard and guaranteed to be inverse-functional within social web application boundaries.

5.5.4 Remarks About Chosen Features

It is important to remember that in this section we outlined an first but not final set of features, result of a partial analysis of existing ontologies and aimed at covering only partially the possible space of properties that can be used to identify an entity of type Person, Location and Organization. Pursuing complete coverage is in principle an endless job as it would require complete knowledge, and would also have probably to deal with many possible inconsistencies. Nevertheless, the approach based on generalization allows to cover a wide set of properties in existing ontologies. Hence, we choose pragmatically to stop extending the set of feature considered, relying on the fact that future application of the method will lead us towards incremental specialization of the vocabulary.

The set of property proposed is quite extensive and provides an initial baseline to future improvements. Future evolutions of the identification ontology types, or specific application scenarios may need further deep analysis related to the level of granularity of the features described so far. Ideally, properties should represent disjunct set of properties. However, this requirement clashes with the need of dealing with data represented at different levels of granularity and precision. For example, it would be great to be able to discern among different level of containment of a location or organization, and not represent both the *administrative level* and the *contained by* features. However, this type of information is available in some sources (e.g. geonames) but not necessarily in others (e.g. dbpedia) where containment relations is often represented with multiple instantiation of the same attribute. We believe that at this stage it is better to include also this type of attributes, relying on the fact that in the future we may be able to discern them more precisely.

Along with the definition of the features for each of the types, we also sketched a brief analysis with respect to possible meta-properties associated to them. In particular, we evaluated each of the features in terms of the meta-properties defined in section 5.4. The analysis of some attributes required the definition of some assumptions forcing some how the assignment of meta-properties. In particular *public institutional id*, *email address* and *website* were forced to be inverse-functional and diachronic, even thou in principle these could be reassigned. We are aware that this choice is prone to cause matching errors, but we believe that doing otherwise would also reduce the possibility of taking positive matching decisions. In a sense, considering time dimension in the assignment of such meta-properties exposed us to some complications we are willing to neglect for the moment. Experimental evaluation will inform us whether this choice has some relative negative consequences.

5.6 Contextual Semantic Harmonization

The heart of a knowledge-based solution is the capability of exploiting the semantic of attributes to take accurate matching decisions using the rules as defined in section 6. To achieve this goal, we decided to rely on a Identification Ontology defining the entity types and their features as described in section 5. One of main goals of such ontology is to provide a point of reference for the harmonization of the semantics of the attributes used in different descriptions. Considering an open and wide environment such as the Web, we have to assume that entities' descriptions are going to be represented differently across heterogeneous sources.

In order to allow the application of matching rules as defined in section 6, the attributes composing such descriptions ought to be mapped towards the Identification Ontology, so that their values can be compared taking into consideration the actual semantics of the attributes. It is clear that this process implies some sort of ontology matching process, aimed at establishing mappings between ontologies and schemas used to shape the collected descriptions and our ontology of reference. Automatic ontology matching is a long studied problem that produced a wide set of solutions [59]. However, the automatic solution of the problem is not the goal of this work, and for the moment we limit our analysis to the existence of the the mappings between the attributes used in the world to describe the considered entity types and Identification Ontology presenting what we call "canonical name" for such attributes.

It is important to remember that we are not dealing with an ideal scenario from this perspective, and even if data are structured with formal ontologies, we have to be careful in the definition of these mappings. In fact, in an open scenario, we have to assume a certain degree of ontological relativity in the interpretation and usage of attributes [122]. Even thou we do not explore the philosophical implications of such relativity, we have to assume that the semantic of attributes defined in ontologies are subject to the interpretation of the people when they instantiate forming what we defined descriptions of entities. This may cause sometimes problems related to the overloading of the semantic of attributes or odd, contextual interpretations. For example, for a matter of convenience, the MusicBrainz[22] schema uses the attribute *begin* and *end* to refer respectively to the date of birth and death of an artist and at foundation and dissolution date of a music band.

For this reasons, we have to conceive a contextualized mapping process allowing to accommodate pragmatic needs related to possibly diverse interpretations of properties defined in ontologies and other types of schemas. The definition of such contextualized mappings follows the intuition behind the definition of Contextual Ontologies presented in [29]. In fact, what we propose is a semantic harmonization process specific for entity matching solution, relying on the existence of mappings defined for this goal. In a sense, in this work we propose to rely on a local interpretation of shared vocabularies and schema to produce local mappings on a local language (i.e. the identification ontology) as defined in [29]. This allows us to conceive mappings as *bridge rules* supporting

[22] `http://musicbrainz.org`

our localized and task specific interpretation of the global semantics of ontologies and schemas used to define the semantics of attributes in the collected descriptions.

With these premises with define $M = \{m \in M | (\alpha, \alpha_c)\}$ as a list of context mappings m defined as a pair of attribute names where α_c represents the canonical name for an attribute defined in the Identification Ontology. Following the syntax used in [29], the mappings m are *bridge rules* of the following types:

- $o : x \xrightarrow{\subseteq} i : y$;
- $o : x \xrightarrow{\supseteq} i : y$;
- $o : x \xrightarrow{\equiv} i : y$;

where x and y are either concepts or properties. Intuitively, these mappings translate the global semantic of the attributes collected and available on the Web into the local semantic defined in the Identification Ontology. The combination of attributes collected and the local mapping generates a context space, where we can interpret the semantic of the attributes relying on the defined contextual mappings, and thus transform the descriptions composed of attributed defined according to a global semantics into descriptions composed of attributes defined according to a local, contextual semantic. We named this process as *semantic harmonization of descriptions*, and define $h : D \times M \rightarrow D$ as the function that takes in input a description $d \in D$ and a set of contextual mappings M and returns the description with the attribute names α replaced by the canonical value α_c defined in the Identification Ontology. To make even more explicit the concept, consider the following pseudocode:

```
harmonize(d in D, M){
   for each a in D{
      if(M.contains(a.n){
         canonical = M.get(a.n);
         a.replace(n, canonical);
      }
   }
}
```

Easing the problem of semantic heterogeneity is a key point in the path towards the definition of a knowledge-based solution relying on rules to take entity matching decision. In fact, in order to compare the attributes considering their semantic, we need first to harmonize the semantic of the properties towards the one defined described in section 5.5.

Chapter 6

Rules for Open Entity Matching

In this section, we aim at formally define syntax and interpretation of matching rules as necessary and sufficient condition to support entity matching decision under the Open World Assumption. In section 6.1 we present a theoretical framework that supports the formulation of matching rules suitable to be employed in as part of a knowledge-based solution to the problem of open entity matching. In section 6.2 we propose a set of tools that support the definition, application and satisfaction of entity matching rules. As mentioned in chapter 4, we aim at constructing rules capturing part of the knowledge used by people in dealing with entity matching problem. This choice implies the adoption of machine learning techniques (i.e. classifiers), whose application may require to perform some operations on the learned rules. For this reason, in sections 6.3 and 6.4 we describe some formal tools to normalize, combine and merge extracted rules. The process of constructing entity matching rules is described in depth in chapter 8.

6.1 Theoretical Foundations

The definition of rules suitable to be applied in a context of the Web under the Open World Assumption requires some sophistication related to the logic underlying their definition, application and satisfaction. In fact, on the one hand we want to avoid to take a negative match decision when positive matching conditions are not satisfied. On the other hand we want to avoid a positive matching decision when a negative matching condition is not satisfied. In a sense, what we need to formalize is something that practically invalidates the classical logic Law of Excluded Middle $MATCH \vee \neg MATCH$ [154]. In fact, under the Open World Assumption, if a positive matching condition is not satisfied, we cannot automatically conclude a negative match. Furthermore, we would also need to invalidate the axiom of Double Negation

Elimination $\neg\neg MATCH \rightarrow MATCH$. In fact, the falsification of a negative matching condition should not lead to a positive match decision. Both the invalidation of the Law of Excluded Middle and the Double Negation Elimination are among the principles underlying the Intuitionistic Logic defined in [146].

However, in Intuitionistic Logic the cases where no decision can be taken are not explicitly formalized, maintaining a boolean valued semantics. Therefore, we choose to represent entity matching rules relying on the Kleene's Three Value Logic [97] as a tool of formalization. The choice is due to the fact that the three value logic can smoothly and explicitly accommodate the unknown matching cases, and provides clear and intuitive interpretation of the usage of logical operators (i.e. connectives) for the definition of rules. Using the Logic of Klenee, we could for example define identification identification rule for a person conveniently represented as a simple conjunction of clauses based on the features defined in the ontology: for example ($Name \wedge Surname \wedge Birthday \wedge Birthplace$), assuming that this combination of attributes leads to unique identification when interpreted in the Web context.

According to the Kleene Logic, the truth value of each of the clauses, and consequently of the whole rule, can be either $TRUE$, $FALSE$ or $UNKNOWN$. The $UNKNOWN$ case allows to accommodate decisions related to the comparison of syntactically heterogeneous attributes. In fact, the solution of any entity matching problem must necessarily pass through some sort of string similarity estimation between the values of features composing a description. Therefore, the truth value of each of the clauses (or atom) composing a matching rule must necessarily pass through the comparison between a string similarity value and a threshold. Traditional boolean operators for comparison (e.g. '¿') would force the falsification of a clause when not satisfied. As a rule would defined as a conjunction of attributes, this would imply also the falsification of the whole rule. For this reason, we need to interpret these operators in a way that would not allow this conclusion to be consistent with our goal.

It is important to remember that in this context, we are not defining rules that have to be applied in a formal logical context, and that we are simply using logic tools to formally describe the business logic that will be then implemented in as a traditional software program ad described in [103, 79]. For example, in order to practically invalidate the Law of Excluded Middle and Double Negation Elimination, we decided to define two complementary set of rules $\mathcal{P}_M$ and $\mathcal{P}_{\neg M}$ leading respectively to positive and negative matching decision. Each set of rules $\mathcal{P}_M$ and $\mathcal{P}_{\neg M}$ can be intuitively interpreted a Kleene's propositional logic formula in disjunctive normal form (DNF) that allow to take different complementary matching decisions. The satisfaction of a single rule (i.e. conjunction of clauses) would imply the satisfaction of the whole formula, and thus support a matching decision. The actual implementation and application of the rules defined in this section will be discussed more in depth in chapters 8 and 9.

For the moment, it is sufficient to remember that our goal is the following: given any set of pairs of descriptions, apply a set of entity matching rules to support positive matching decision when a positive matching rule is satisfied, support negative matching decision when a non-match rule is satisfied, and unknown in any other case. The following paragraphs are aimed the formalization of the tools necessary to implement this intuitive logic described above into a system of rules. Notice that for this formalization, we do not need to distinguish between positive and negative matching rules, therefore we simply consider $\mathcal{P}$ as a generic set of all rules.

6.2 Rules Definition, Application and Satisfaction

Lets define a matching rule as a conjunction of rule atoms $\theta < \alpha, o, t >$ where α is a feature defined in the Identification Ontology, $o \in O$ is an operator among $O : \{=, >, <, \leq, \geq\}$ interpreted in the context of the Kleene logic, and t is a similarity threshold in the range $[0, 1]$ to be used as term of comparison for the satisfaction of the atom according to the operator o. More formally, a rule $\rho \in \mathcal{P}$ can be defined as:

$$\rho = \bigwedge_{i=0}^{n} \theta_i \tag{6.1}$$

A rule ρ *applies* to a pairs of descriptions $d_1, d_2 \in D$ if the intersection of the attributes α composing the descriptions d_1 and d_2 contains all the features composing ρ. We need then to formalize a few simple functions that would allow us to formally define the function $apply_\rho$ that would allow us to estimate whether a rule can be applied to a pair of descriptions. First of all we need a function that given a rule, extracts the features type composing it. More formally, lets define $\delta_\rho : \mathcal{P} \to A$ as the function that given a rule, returns the set of attributes composing it:

$$\delta_\rho(\rho) : \{\alpha \in A | \exists \theta \in \rho \wedge \alpha \in \theta\}. \tag{6.2}$$

Then we need a function that given a description extracts the feature composing it. Using the metaphor of the fingerprint analysis described in the beginning of chapter 4, this step corresponds to individualizing the features are present on a fingerprint, or in this case, in a description. Lets then define also the function $\delta_d : D \to A$ as the function that given a description $d \in D$ returns the set of attribute names α composing it:

$$\delta_d(d) : \{\alpha \in A | \exists a \in d \wedge \alpha \in a\}. \tag{6.3}$$

Given these two functions, δ_ρ and δ_D, we can now define the function that verifies whether a rule can be applied to a pair of descriptions. Intuitively, given these two functions, we can simply consider the set of attributes result of these functions applied on the descriptions and the rule, and if the attribute composing a rule obtained applying δ_ρ is a proper subset of the intersection between attributes in common between the two compared descriptions, then we can assert that the rule *applies*. More formally, lets define $apply_\rho : \mathcal{P} \times D \times D \rightarrow B$ as the boolean function that taken a rule and two descriptions verifies whether the rule applies for comparing the two descriptions:

$$apply_\rho(\rho, d_1, d_2) : \begin{cases} true, & \text{if } \delta_\rho(\rho) \subseteq (\delta_d(d_1) \cap \delta_d(d_2)) \\ false, & \text{otherwise} \end{cases} \tag{6.4}$$

Notice that the process of selecting the attributes that apply for the application of a rule allow also to ease the problem of over-specification described in chapter 2. In fact, selecting only relevant attributes for comparison, we avoid comparing special purpose attributes which are not interpretable in a global context.

Now that we defined when a rule applies to a pair of descriptions, we need to complete the set of tools and define the function that decides when a matching rule is satisfied when comparing two descriptions. In particular a rule $\rho \in \mathcal{P}$ is satisfied when:

1. it can be applied to a pair of descriptions d_1 and d_2;

2. all the atoms $\theta_i \in \rho$ are satisfied;

So far, we defined the tools supporting an analysis about whether a rule can be applied to a set of descriptions. The next step is to formally define the functions that support decisions about whether a single atom is satisfied. This servers as tool to estimate whether a rule is satisfied or not. An atom θ is satisfied if and only if among all the values of the attributes $a_i \in d_1$ and $a_j \in d_2$ of type α with $\alpha \in \theta$, the comparison of the values $v_i \in a_i$ and $v_j \in a_j$ according to some string similarity metrics produces a score satisfying a the Kleene operator $o \in \theta$ with respect to the threshold $t \in \theta$. Hence, we first have to define the function

$$\kappa : \Omega \times [s_1, ..., s_n] \times [s_1, ..., s_m] \rightarrow \Re \in [0, 1] \tag{6.5}$$

that given to list of strings $s_1, ..., s_n$ and $s_1, ..., s_m$ with $s_i \in \mathcal{S}$ representing the values of semantically equivalent features, returns a similarity measure between 0 and 1 according to a similarity metric $\omega \in \Omega$. In this context, a function $\omega \in \Omega$ represents a string similarity metric possibly among the one presented in section 3.3 of the state of the art. The function κ can be interpreted as some sort of second order function, abstracting the application of any string similarity metric selected, and embedding a process of comparison of semantically equivalent features. Given a κ function, we need now to define the function that decides whether a rule atom θ is satisfied or not. Reminding that any atom θ is defined as the tuple $< \alpha, o, t >$, we need to define the function that given the result of a comparison obtained by the application of κ, verifies whether the similarity score of the two strings satisfies the operator o. Keeping in mind our primary goal of not falsifying any matching rule that cannot be completely satisfied, lets define $satisfy_o : \Re \times \Re \times O \rightarrow \{TRUE, UNKNWON\}$ as the function that given two positive real numbers $r_1, r_2 \in \Re$, and a comparison Kleene operator $o \in O$ returns the result of the test:

$$satisfy_o(r_1, r_2, o) : \begin{cases} true, & \text{if } o(r_1, r_2) = true \\ unknown, & \text{otherwise} \end{cases} \tag{6.6}$$

Also the $satisfy_o$ function is a function of second order, that allows the application of different operators according to need. At this point, given a function κ to compute similarity between two strings, and function $satisfy_o$ to verify whether a pairs of real number satisfies a comparison Kleene operator, we can define a function $satisfy_\Theta$ that, given two list of values of a feature f, verifies whether a rule atom is satisfied. More formally, lets define $satisfy_\theta : \Theta \times [v_{f1}, ..., v_{fn}] \times [v_{f1}, ..., v_{fm}] \times \Omega \rightarrow \{TRUE, UNKNWON\}$ as the function:

$$satisfy_\theta(\theta, V(a_1), V(a_2), \omega) \begin{cases} true, & \begin{aligned} &\text{if } satisfy_o(\kappa(\omega, V(f_1), \\ &Vf_2), t, o) = true \text{ and} \\ &\qquad\qquad t, o \in \theta \end{aligned} \\ unknown, & \text{otherwise} \end{cases} \tag{6.7}$$

assuming that $(\alpha_\theta \in \theta \wedge \alpha_1 \in a_1 \wedge \alpha_2 \in a_2 \wedge \alpha_\theta = \alpha_1 = \alpha_2)$. Namely, assuming that the attribute type of the feature $f_1, f2$ and θ as the same. Notice that $V(f)$ is compact syntactic representation of the the vector of values of a feature f in a description.

Now that we defined the function $satisfy_\theta$, we can proceed defining the function that verifies whether a rule is satisfied given two descriptions. Intuitively, the function now must simply verify that given a pair of descriptions, when compared, the atoms of a rule are satisfied. Then we can define $satisfy_\rho : \mathcal{P} \times D \times D \rightarrow B$ as the function that takes in input a rule $\rho \in \mathcal{P}$ and two descriptions $d_1 \in D$ and $d_2 \in D$:

$$satisfy_\rho(\rho, d_1, d_2) : \begin{cases} true, & \text{if } \begin{aligned} &\forall \theta \in \rho, \exists f_1 \in d_1, f_2 \in d_2 \\ &\wedge satisfy_\theta(\theta, V(f_1), V(f_2)) \end{aligned} \\ unknown & \text{otherwise} \end{cases} \tag{6.8}$$

assuming that $apply_\rho(\rho, d_1, d_2)$ holds.

In this section we formally defined the tools necessary to apply and satisfy a matching rule under the Open World Assumption. These tools are generic, and apply both to positive and negative matching rules.

6.3 Rules Normalization

In previous section we formally defined entity matching rules, and some tools to apply them and verify their satisfaction. This section presents a set of normalization operation aimed at reducing possible inconsistencies or counterintuitive results of bottom up rules extraction with respect to the formalization given. Each of the following subsections presents normalization operation aiming at fixing specific types of odds we can find in learned rules in order to normalize them and make them consistent with respect to the formalization defined. The normalization steps take into consideration single atoms and sets of atoms to produce normalized version of the rules. Some of the normalization processes outlined in the following are the result of intuition and heuristics. Therefore the impact of such heuristic will have to be empirically evaluated through experiments on real data.

6.3.1 Atom Operator Normalization

Given a positive matching rule, it seems counterintuitive to find an atom with a comparison operator stating that a value must be below a similarity threshold $e.g.description < 0.6$. In fact, this would imply that a positive matching rule could be satisfied only when two attributes are necessarily different. This type of atom can be extracted from datasets presenting very heterogeneous values for the same attribute types. Indeed, attributes such as "description" could contain perfectly matching values for negative matching samples, and at the same time, and very poorly matching values for positive matching samples. This can be due to the fact that negative matching description contain short values for some attributes, whereas positive matching samples compare strings of very different length and consistence (e.g. a whole paragraph compared with a single sentence). These samples might create some troubles in the learning process.

An atom rule is consistent if coherent with the matching decision supported. If the rule is a positive matching rule, and the operator $o \in \theta$ is among $\{>, \geq\}$, the atom is coherent, otherwise it is not. If the rule is a negative matching rule, and the operator $o \in \theta$ is among $\{<, \leq\}$, the atom is coherent, otherwise is not.

For the reasons above, we need to normalize the rules to correct the inconsistent operator extracted. There are two options:

1. remove the atom from the rule;
2. normalize the operator according the rule decision;

The first option is more radical, as the removal of an atom from the rule would imply not considering that attribute for matching decision as clearly it is not reliable to take 'similarity metrics'-based matching decision. An alternative interpretation would be that this attribute is not relevant to take matching decisions, in combinations with the others. If we apply this principle consistently, given the high degree of heterogeneity and noise we can possibly find the data, we are likely to define shorter rules. Discarding atoms could be useful to avoid considering noisy attributes in the entity matching decision process. More formally, let's defined the normalization function $\eta_R : \mathcal{P} \rightarrow \mathcal{P}$ that given a rule ρ, removes atoms θ inconsistent with the decision supported by the rule.

The second option is more conservative, as does not discard the fact that the classifier considered that attribute as discriminant for taking precise matching decision. More formally, let's defined the normalization function $\eta_C : \mathcal{P} \rightarrow \mathcal{P}$ that given a rule ρ, changes the operator of the inconsistent atoms θ as incoherent with the decision supported by the rule. An atom rule is consistent if coherent with the matching decision supported. If the rule is a positive matching rule, and the operator $o \in \theta$ is among $\{<, \leq\}$, the atom is should be changed in $\geq$. If the rule is a negative matching rule, and the operator $o \in \theta$ is among $\{>, \geq\}$, the atom operator should be changed in $\leq$.

Changing the sign of the operator according to the rule decision could help in building robust and precise rules. However, if the classifier split the dataset based on a negative similarity threshold, being conservative we risk to have a robust rule that may not get satisfied matching noisy descriptions. Whereas removing the inconsistent atom, we risk to create short rules that produce unreliable matching decisions. As previously mentioned, we need to experimentally evaluate the impact of both options.

6.3.2 Transitive Operator Normalization

Previous normalization steps on similarity operators could produce rules ρ that could contain several atoms θ with the same feature f but different operators and thresholds, e.g. $name > 0.3 \wedge name > 0.5 \wedge name > 0.8 \wedge birthday > 0.8 \wedge birthplace > 0.9$. This type of rule demands redundant attribute evaluation to establish its satisfiability. That is, the attribute *name* in the example would have to satisfy 3 atoms on the same value. However, intuitively the three comparisons become useless considering the transitivity property of the operator $>$. Indeed, if the name comparison is tested to be larger than 0.8, than it is also implicitly larger than 0.3 and 0.5.

To avoid this redundant evaluations, we propose to normalize rules according to the transitivity property of the operator used in the single atoms.

Given a rule ρ with more than one atom θ containing a feature f with the compatible operator o, $|\{\theta \in \rho | f, o \in \theta\}| > 1$, we need to define a normalization function η_T to perform the transitive reduction of the rule, and thus producing a minimal representation of the rule. The transitive reduction of rule $\rho \in \mathscr{P}$ on a relation (or operator) r can be built simply choosing greedily among the atoms with same property and compatible operator, the one with maximum threshold for operators $\{>, \geq\}$, and the one with minimum threshold for operators $\{<, \leq\}$. More formally, lets define the function $\pi_\theta : \mathscr{P} \times \{o_1, ..., o_m\} \times F \to \Theta[...]$ that given a rule and set of operators and a feature selects the atoms containing one of the operators and the feature. Lets define the function $norm_\theta : \Theta[] \to \Theta$ as the function that given an array of atoms selects the one with minimum ($minT$) or maximum threshold ($maxT$) according to the operator:

$$norm_\theta([\theta_1, ..., \theta_n]) : \begin{cases} minT([\theta_1, ..., \theta_n]), & \text{if } \forall i(\theta_i.o \in \{<, \leq\}) \\ maxT([\theta_1, ..., \theta_n]), & \text{if } \forall i(\theta_i.o \in \{>, \geq\}) \end{cases} \tag{6.9}$$

Therefore, the transitive reduction produces rules preserving the atom with the higher similarity threshold for positive matching rules, and the atom with the lower similarity threshold for the negative matching rules. This operation does not affect the actual effectiveness of a rule in supporting matching decision, but rather simplifies the computation of its possible satisfaction.

6.4 Rules merging

Previous normalization steps, including transitive reduction on Klenee operators may produce rules whose application is subsumed by others. In this case, in order to optimize the set of rules and enforce their soundness we may need to merge these rules. It is important to remind that the main requirement driving the definition of rules is soundness. Namely, the rules that are defined must lead to precise and reliable decisions. Given the problems related by imprecise matching methods producing unreliable *owl:sameAs* statements, we affirm that completeness is not so relevant, meaning that it is not essential to define rules covering all possible matching cases. Intuitively, there are two cases that could necessitate of a merging of a rule:

- one rule is subsumed by the other;

- two rules are equivalent;

A formal definitions of rules subsumption is presented in section 6.4.1. A merging principle based on rules subsumption is presented in section 6.4.2, and finally in section 6.4.4 merging of equivalent rules is treated.

6.4.1 Rules subsumption

We define $\rho_2 \sqsubseteq_\rho \rho_1$, read "a rule ρ_1 subsumes another rule ρ_2, when the set of pairs of descriptions to which ρ_2 applies is a proper subset or equivalent to the set of pairs of descriptions to which ρ_1 applies. More formally, define D_ρ as the set of descriptions to which a rule ρ applies:

$$D_\rho(\rho, D) = \{d_1, d_2 \in D | apply_\rho(\rho, d_1, d_2,)\} \tag{6.10}$$

with $apply_\rho$ as defined in equation (6.4) at page 102. We can now define rules subsumption in terms of sets of descriptions to which a rule applies. That is, if a rules ρ_2 applies to a subset of descriptions to which applies a rule ρ_1, then we say that rule ρ_2 subsumes rule ρ_1. More formally:

$$\rho_2 \sqsubseteq_\rho \rho_1 \forall D(\Leftrightarrow D_\rho(\rho_2, D) \subseteq D_\rho(\rho_1, D)). \tag{6.11}$$

This pragmatic definition of rules subsumption allows us to formalize a hierarchy of rules with respect to their level of generality. At the top of this hierarchy it is possible to find the most general rule, i.e. the empty rule ρ_T such that given any set of pairs of descriptions $d_1, d_2 \in D$, the rule would applies to all the descriptions. That is $\forall D, D_\rho(\rho_T, D) = D$. At the bottom of the hierarchy the most specific rule ρ_B such that it does not apply to any pair of descriptions $\forall D, D_\rho(\rho_B, D) = \emptyset$. Furthermore, we define that two rules are equivalent when these rules subsume each other. More formally,

$$\rho_2 \sqsubseteq_\rho \rho_1 \wedge \rho_1 \sqsubseteq_\rho \rho_2 \Leftrightarrow \rho_2 \equiv_\rho \rho_1. \tag{6.12}$$

With equations (6.11) and (6.12), we formally defined the notion of rules subsumption. We can now use this notion to formally define a merging process, as outlined in the following section.

6.4.2 Merging ρ-subsumed rules

Given a hierarchy of rules based on the rules subsumption relation as defined in section 6.4.1, we have a principle to define a merging function supporting the definition of robust and specific or very general matching rules. Following the principle that soundness of matching rules is more important than completeness, in this section we propose to merge the matching rules in favor of the most specific rules. However, given the definition of subsumption, we can easily produce generic rules.

Intuitively, the definition of sound rules implies choosing that the longer rule will always be maintained, and the shorter one would be merged into the longer one. More formally, given the rules $\rho_1 = a \wedge b \wedge c$ and $\rho_2 = a \wedge b \wedge c \wedge d$, then $\rho_1 \sqsubseteq \rho_2$, as there may exist a description containing only the features a, b, and c to which rule ρ_2 does not apply. Therefore, we need a function, that given two rules, decides which one has to be merged in the other. We call *pivot rule* the rule persisting, and *merged rule*, the rule merged in the pivot. With the purpose of choosing the pivot between two rules, we define the function $\mu_r : \mathcal{P} \times \mathcal{P} \to \mathcal{P}$, that given two rules chooses as pivot the more restrictive based on subsumption:

$$\mu_r(\rho_1, \rho_2) = \begin{cases} \rho_2, & \text{if } \rho_2 \sqsubseteq_\rho \rho_1 \\ \rho_1, & \text{if } \rho_1 \sqsubseteq_\rho \rho_2 \end{cases} \tag{6.13}$$

Conversely, lets define $\mu_u : \mathcal{P} \times \mathcal{P} \to \mathcal{P}$, that given two rules chooses as pivot the more unrestrictive based on subsumption:

$$\mu_u(\rho_1, \rho_2) = \begin{cases} \rho_1, & \text{if } \rho_2 \sqsubseteq_\rho \rho_1 \\ \rho_2, & \text{if } \rho_1 \sqsubseteq_\rho \rho_2 \end{cases} \tag{6.14}$$

Both restrictive and unrestrictive merging pivot selector assume that the merged rules have the same classification target. That is, positive matching rules are merged with positive matching rules only, and negative matching rules are merged with negative matching rules only, for the moment. Notice that if the rules are ρ-equivalent, i.e. $\rho_1 \equiv_\rho \rho_2$, the selection of the pivot does not matter.

In the merging process, once we selected the pivot according either using the μ_r or μ_u function, we need to perform the actual merging of the rules, dealing with rule atoms presenting different operators and thresholds. Reminding that the principle driving the extraction of rules is soundness, equivalent rules with different similarity thresholds should be merged to be more restrictive. That is, two mergeable positive matching rules presenting atoms with different similarity threshold should be merged selecting the higher threshold. Conversely, negative matching rules presenting atom with different similarity should be merged selecting the lower similarity threshold. Thereby, a pivot selection function for atoms in merged positive matching rules $\mu_{\theta\geq} : \Theta \times \Theta \to \Theta$ would be:

$$\mu_{\theta\geq}(\theta_1, \theta_2) : \begin{cases} \theta_1, & \text{if } \theta_1 f = \theta_2 f \wedge \theta_1 t \geq \theta_2 \\ \theta_2, & \text{if } \theta_1 f = \theta_2 f \wedge \theta_2 t \geq \theta_1 \end{cases} \tag{6.15}$$

Whereas, a pivot selection function for negative matching rules $\mu_{\theta\leq} : \Theta \times \Theta \rightarrow \Theta$ would be:

$$\mu_{\theta\leq}(\theta_1, \theta_2) : \begin{cases} \theta_1, & \text{if } \theta_1 f = \theta_2 f \wedge \theta_1 t \leq \theta_2 \\ \theta_2, & \text{if } \theta_1 f = \theta_2 f \wedge \theta_2 t \leq \theta_1 \end{cases} \tag{6.16}$$

At this point, we formally defined tools for selection of pivot rules in merging process, and tools for the selection of atoms to complete the merging process. Intuitively, the merging of two rules now consists in simply selecting the pivot one using either μ_r or μ_u, and for each of the atoms of the pivot rule, select the one with the more restrictive threshold.

Keeping in mind the overall goal of extracting entity matching rules that could be employed in a decision process suitable for open entity matching, we have also to deal with the fact that different rules might define threshold for the satisfaction of the rule atoms with the same feature. In fact, different datasets could present syntactically very heterogeneous values for the same attribute type. For example, the attribute "description" is suitable to present information in textual form of different length, attributes of type "date" could be represented in different formats, attribute "name" could be represented with, or without title or middle name, etc. Thereby, the similarity thresholds on different datasets could change considerably for the same feature. This aspect may be mitigated by filtering out attributes particularly heterogeneous. However, converging perfectly on a single similarity threshold is unlikely to happen, and it does not make sense to consider different thresholds for the same feature in different rules.

6.4.3 Similarity Thresholds Normalization

In previous sections, we defined normalization processes, aimed at normalizing single rules inconsistencies, and formally defining merging condition and process. However, it is reasonable to assume the possibility of having rules that cannot be merged as the do not apply to the same of samples, but presenting common rule atoms with different similarity thresholds for the same feature. However, in a knowledge based solution the semantic of the attributes is clear, and the satisfaction of each atom has to be interpreted as a Kleene operator behaving consistently in different rules. Intuitively, if we learn from data that two names to match should have a minimal similarity of t, it does not make sense that in another rule we considers the same attribute with similarity threshold t_1 and $t_1 > t$ or $t_1 < t$. Given two descriptions, the decision about the satisfaction of a rule atom should be uniform across all rules classifying the same class. Therefore, we want to normalize all the similarity thresholds for the same feature in different rules sharing classification purposes. At this point, we assume that the rules have been already normalized and merged as discussed in sections 6.3 and 6.4.

We need first to select the rule atoms shared among the rules. To do so, we can simply process the rules using the features defined in the identification ontology, and extract for each of them all the atoms presenting that feature. More formally, lets define the function $\pi_{\mathscr{P}[...]} : \mathscr{P}[...] \times \{o_1, ..., o_m\} \times F \rightarrow \Theta[...]$ that given a list of rules and set of operators and a feature returns all the atoms containing one of the operators and the feature.

We now need to define simple operators that select the minimum and maximum thresholds among the set of rule atoms selected using the function $\pi_{\mathscr{P}[...]}$. More formally, lets define $minT : \Theta[...] \rightarrow \Theta$ as function that given an array of atoms selects the one with minimum threshold. Intuitively, this function can be defined in terms of iterative application of the $\mu_{\theta\leq}$ on pairs of atoms.

Similarly, let's define $maxT : \Theta[...] \times O \rightarrow \Theta$ a function that given an array of atoms selects the one with maximum threshold. Intuitively, this function can be defined in terms of iterative application of the $\mu_{\theta\geq}$ on pairs of atoms.

So far, we defined tools for selecting atoms with maximum and minimum thresholds, given a feature and set of operators. We now need a function that replaces such atoms in all rules, so that similarity thresholds can be normalized across all defined rules. Let's define $replace : \mathscr{P} \times \Theta \rightarrow \mathscr{P}$ the function that given a rule ρ and an atom $\theta < \alpha, o, t >$ where α is a feature, o is an operator among $O\,\{=, >, <, \leq, \geq\}$ and t is a similarity threshold in the range $[0, 1]$, replaces the atoms $\theta_i \in \rho$ such that $\theta_i.f = \theta_p.f\theta_i.o = \theta_p.o$ with θ_p.

We now have the instruments to define a first set of simple cross-rule similarity threshold normalizers aimed at the production of a set of rules presenting uniform thresholds for rule atoms supporting the same decision. The selection of maximum and minimum threshold implies the definition of more relaxed or conservative rules according to their final goal. In the following we present possible combinations of threshold normalization choices:

- *Conservative Match and Conservative Non Match.* Aiming at defining sound rules, one possibility is to make the rules more conservative and to minimize the set of descriptions pairs satisfying the rule and consequently maximizing the set of rules classified as *DONTKNOW*. This can be done by selecting the maximum similarity threshold for each feature among all positive matching rules, and selecting of minimum threshold for each feature among all negative matching rules. This approach is conservative towards both MATCH and NONMATCH classification, and it is noted as *CC* (Conservative-Conservative). More formally, for each positive matching rule ρ in the list of rules $\mathscr{P}[]$, we apply $replace(\rho, maxT(pi_{\mathscr{P}[...],\{>,\geq\}}))$, and for each negative matching rule ρ in the list of rules $\mathscr{P}[]$, we apply $replace(\rho, minT(pi_{\mathscr{P}[...],\{<,\leq\}}))$.

- *Conservative Match and Relaxed Non Match.* Aiming at maximizing the set of pairs of descriptions satisfying negative matching rules, being conservative about positive matching rules, we can select the maximum threshold for each features in negative matching rules. This approach is conservative towards MATCH classification but more relaxed on NONMATCH as the set of pairs of description possibly satisfying the rule is enlarged, and it is noted as CR (Conservative-Relaxed). More formally, for each positive matching rule ρ in the list of rules $\mathscr{P}[...]$, we apply $replace(\rho, maxT(\pi_{\mathscr{P}[...]}, \{>, \geq\}))$, and for each negative matching rule ρ in the list of rules $\mathscr{P}[...]$, we apply $replace(\rho, maxT(\pi_{\mathscr{P}[...]}, \{<, \leq\}))$.

- *Relaxed Match and Conservative Non Match.* Aiming at maximizing the set of pairs of descriptions satisfying positive matching rules, being conservative about negative matching rules, we can select the minimum threshold for each features in positive matching rules. This approach is conservative towards NONMATCH classification but more relaxed on MATCH as the set of pairs of description possibly satisfying the rule is enlarged, and it is noted as RC (Relaxed-Conservative). More formally, for each positive matching rule ρ in the list of rules $\mathscr{P}[...]$, we apply $replace(\rho, minT(\pi_{\mathscr{P}[...]}, \{>, \geq\}))$, and for each negative matching rule ρ in the list of rules $\mathscr{P}[...]$, we apply $replace(\rho, minT(\pi_{\mathscr{P}[...]}, \{<, \leq\}))$.

- *Relaxed Match and Relaxed Non Match.* Aiming at improving completeness of rules, one possibility is to make the rules less conservative and to maximize the set of descriptions pairs satisfying each rule and consequently minimizing the set of rules classified as $DONTKNOW$ by selecting the minimum similarity threshold for each feature among all positive matching rules, and selecting of maximum threshold for each feature among all negative matching rules. This approach is relaxed towards both MATCH and NONMATCH classification, and it is noted as RR (Relaxed-Relaxed). More formally, for each positive matching rule ρ in the list of rules $\mathscr{P}[...]$, we apply $replace(\rho, minT(\pi_{\mathscr{P}[...],\{>,\geq\}}))$, and for each negative matching rule ρ in the list of rules $\mathscr{P}[...]$, we apply $replace(\rho, maxT(\pi_{\mathscr{P}[...]}, \{<, \leq\}))$.

The effects of such normalizations have to experimentally evaluated.

6.4.4 Rules Merging process

The merging function based on rule subsumption defines a partial order in a set of rules, thereby the rules merging process can be done simply iterating on list of rules until no further merging is possible among the rules. Consider the following pseudocode to describe the iterative normalization process.

```
process(List rules){
  List normalized = normalize(rules);
  int normalizedSize = normalized.size;
  int mergedSize = 0;
  while(mergedSize != normalizedSize){
     normalizedSize = normalized.size;
     List merged = {};
     Set m = {};
     for(int i=0; i<normalizedSize-1;i++){
         for(int j=i+1; i<normalizedSize; j++){
            if(!m.contains(i) && !m.contains(j)){
                if(mergeable(normalized[i], normalized[j]){
                    merged.add(merge(normalized[i], normalized[j]);
                    m.add(i);
                    m.add(j);
                }
            }
         }
     }
     for(int i=0; i<normalizedSize;i++){
         if(!m.contains(i)){
             merged.add(normalized[i]);
         }
     }
     normalized = merged;
     mergedSize = merged.size;
  }
}
```

The *normalize* method takes the rules and applies a set of normalization steps as described in section 6.3. The *mergeable* method is a boolean method that taken in input two rules applies either μ_r or μ_u to decide whether a pair of rules can be merged, and which one is the pivot.

6.4.5 Defining Rules Class Hierarchy

In section 6.1 we framed the definition of the rules to support matching decisions consistently with the Three Value logic of Kleene. It is clear that $MATCH$ and $\not{M}ATCH$ rules, even if they are equivalent (i.e. they apply to the same set of rules), they have to be disjoint and considered separately. Different analysis applies to $DONTKNOW$ classification rules. The normalization and merging steps defined so far may end up in producing equivalent matching rules belonging to $MATCH$ and $DONTKNOW$ classes or $\neg MATCH$ and $DONTKNOW$ classes. If two equivalent rules produce a different matching decision, we should be able to decide which rule should be applied, besides the order of application of the rules. However, it is important to consider also similarity thresholds and operators in this case. In fact, we defined normalization operations to remove or fix incoherent operators for $MATCH$ and $\not{M}ATCH$ classification rules, but did not deal with $DONTKNOW$ cases. This does not allow us to apply simple ρ-equivalence to define a merging process, but forces us to rely on more strict rules equality. Namely, two roles are ρ-equal, if the they present they satisfy exactly the same set set of descriptions pairs, for any descriptions set. This implies that the rules are a conjunction of exactly the same atoms. Therefore, we propose to define also hierarchy of matching classes, so that the selection process of ρ-equal rules belonging to different classes can be formally driven. Intuitively, if we have two ρ-equal rules, one producing a $MATCH$ decision, and the other producing a Dont Know decision, we should conservatively choose the $DONTKNOW$ decision, as $MATCH$ decision may be imprecise. We can then formalize the hierarchy it in the following way:

$$MATCH \sqsubseteq DONTKNOW \tag{6.17}$$

Similarly, if we have two equivalent rules that classify a NonMatch on in one case and Dont Know on the other, then the merged rule should be a DontKnow as the NonMatch decision could be too radical, i.e. imprecise. We can then formalize it in the following way

$$\neg MATCH \sqsubseteq DONTKNOW \tag{6.18}$$

.

Thereby, a the selection process based on such hierarchy $\mu_c : \mathcal{P} \times \mathcal{P} \rightarrow \mathcal{P}$ would be:

$$\mu_c(\rho_1, \rho_2) = \begin{cases} \rho_1, & \begin{array}{r} \text{if } \rho_2 \notin DONTKNOW \wedge \\ \rho_1 \in DONTKNOW \wedge \\ \rho_1 \equiv_{rho} \rho_2 \end{array} \\ \rho_2, & \begin{array}{r} \text{if } \rho_1 \notin DONTKNOW \wedge \\ \rho_2 \in DONTKNOW \wedge \\ \rho_1 \equiv_{rho} \rho_2 \end{array} \end{cases} \tag{6.19}$$

We believe that the definition of such hierarchy guarantees the definition of sound rules, removing possible inconsistency due to partial normalization and processes.

6.5 Remarks about Rules for Open Entity Matching

In this section we formally defined the matching rules and some tools to verify their application and satisfaction. We also formally framed the application of these rules under the Open World Assumption, and modeled the satisfaction of rules around Kleene

Three Value logic. Reminding the our goal is not to implement the rules in a formal system, we believe we provided a sound formalization of the tools we need to integrate and apply these rules in a more complex software program providing a reliable rule-based solution to the open entity matching problem. It is therefore clear that the construction of matching rules suitable for open entity matching is a key point of the solution proposed in this work. This aspect is treated in details in the chapter 8. However, before applying any rule, we need to deal with the problem of semantic heterogeneity affecting the descriptions available on the (Semantic) Web. In fact, the matching rules defined in this section are expressed in terms of the Identification Ontology. Therefore, we necessarily have to harmonize the semantic of the descriptions using the features defined in it in order to be able to apply them. The solution to this problem is presented in section 5.6.

In this chapter, we formally outlined the principles and some theoretical grounding of the knowledge-based solution we propose to the entity matching problem. In chapters 8 and 9 we are going to further explore some aspects treated in this chapter, and mostly going to describe a first implementation of the knowledge-based solution. In chapter 10, we are then going to experimentally evaluate the impact of the many heuristic intuitions we outlined in this chapter aimed at proposing solutions to inherent problems.

Part III

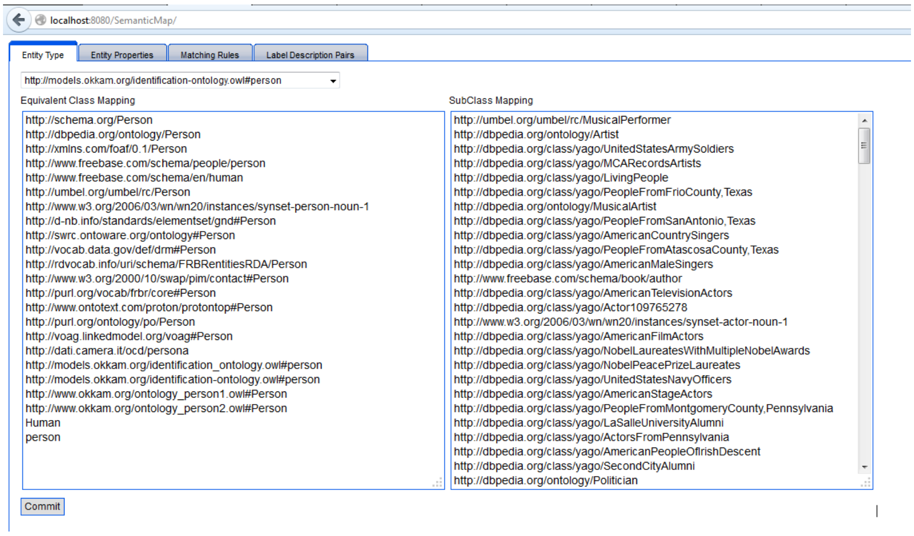
Implementation and Evaluation

Chapter 7

Semantic and Structural Harmonization

A necessary condition for the application of any rule defined in terms of the ontology is to harmonize the semantic of the attributes used in the description defining contextual mappings towards the identification ontology. The automatic definition/discovery of such mappings is not tackled particularly in this work, relying on the existence of a set of mapping regardless their provenance. Nevertheless, in order to bootstrap the knowledge-based solution we proposed in this work, it is necessary to define a first set of mappings. In particular, in section 7.1 we preset list of mappings to harmonize known entity types, whereas in section 7.2 we describe more in details the mappings related to features defined for the types in the identification ontology. Besides this initial batch of manually defined mappings, we foresee the adoption of some semi-supervised method to solve this problem. In alternative, we can also simply assume that before any application of the solution proposed in this work, a human experts provides the necessary mappings to complete the process. With respect to the last point, tools such as Open Refine[1] and Karma [144] support the manual and automatic definition of these mappings dealing with different types of data sources.

The problem of structural harmonization is only partially treated in this work. In particular we focused on few attribute types such as names, dates and geo-coordinates, defining a set of transformation functions that will be extended in the future. Details about this process are presented in section 7.3.



localhost:8080/SemanticMap/

http://models.okkam.org/identification-ontology.owl#person

Equivalent Class Mapping

```
http://schema.org/Person
http://dbpedia.org/ontology/Person
http://xmlns.com/foaf/0.1/Person
http://www.freebase.com/schema/people/person
http://www.freebase.com/schema/en/human
http://umbel.org/umbel/rc/Person
http://www.w3.org/2006/03/wn/wn20/instances/synset-person-noun-1
http://d-nb.info/standards/elementset/gnd#Person
http://swrc.ontoware.org/ontology#Person
http://vocab.data.gov/def/drm#Person
http://rdvocab.info/uri/schema/FRBRentitiesRDA/Person
http://www.w3.org/2000/10/swap/pim/contact#Person
http://purl.org/vocab/frbr/core#Person
http://www.ontotext.com/proton/protontop#Person
http://purl.org/ontology/po/Person
http://voag.linkedmodel.org/voag#Person
http://dati.camera.it/ocd/persona
http://models.okkam.org/identification_ontology.owl#person
http://models.okkam.org/identification-ontology.owl#person
http://www.okkam.org/ontology_person1.owl#Person
http://www.okkam.org/ontology_person2.owl#Person
Human
person
```

SubClass Mapping

```
http://umbel.org/umbel/rc/MusicalPerformer
http://dbpedia.org/ontology/Artist
http://dbpedia.org/class/yago/UnitedStatesArmySoldiers
http://dbpedia.org/class/yago/MCARecordsArtists
http://dbpedia.org/class/yago/LivingPeople
http://dbpedia.org/class/yago/PeopleFromFrioCounty,Texas
http://dbpedia.org/ontology/MusicalArtist
http://dbpedia.org/class/yago/PeopleFromSanAntonio,Texas
http://dbpedia.org/class/yago/AmericanCountrySingers
http://dbpedia.org/class/yago/PeopleFromAtascosaCounty,Texas
http://dbpedia.org/class/yago/AmericanMaleSingers
http://www.freebase.com/schema/book/author
http://dbpedia.org/class/yago/AmericanTelevisionActors
http://dbpedia.org/class/yago/Actor109765278
http://www.w3.org/2006/03/wn/wn20/instances/synset-actor-noun-1
http://dbpedia.org/class/yago/AmericanFilmActors
http://dbpedia.org/class/yago/NobelLaureatesWithMultipleNobelAwards
http://dbpedia.org/class/yago/NobelPeacePrizeLaureates
http://dbpedia.org/class/yago/UnitedStatesNavyOfficers
http://dbpedia.org/class/yago/AmericanStageActors
http://dbpedia.org/class/yago/PeopleFromMontgomeryCounty,Pennsylvania
http://dbpedia.org/class/yago/LaSalleUniversityAlumni
http://dbpedia.org/class/yago/ActorsFromPennsylvania
http://dbpedia.org/class/yago/AmericanPeopleOfIrishDescent
http://dbpedia.org/class/yago/SecondCityAlumni
http://dbpedia.org/ontology/Politician
```

Figure 7.1: A view of the Semantic Map application

7.1 Entity Type Harmonization

From a practical perspective, the harmonization of the semantic of both entity types and attributes is fairly simple, as it assumes the existence and availability of a list of contextual mappings both for the entity types considered, and also for the attributes associated to them. Contextual mappings to known existing types (or concepts) are going to be used to harmonize the type mentioned in the descriptions, replacing it with the canonical type declared in the Identification Ontology. We are aware that some of the entities might not present any information related to the entity type, or that mapping might not always be available. In this context we do not explore automatic methods for guessing the type of the entity given a description (e.g. [5, 6]), and we postpone it to future work. Discerning the type of the entity to which a description is referring to is essential to the application of any knowledge-based solution. Thereby, when entity type harmonization fails the matching process cannot be completed.

In order to ease the problem of contextual mapping collection and maintenance, we implemented a first simple web application supporting the management of this task. We temporarily named this application *Semantic Map*, as it supports the extension and maintenance of the contextual mappings towards the Identification Ontology. In figure 7.1 a view of the application in the part that allows the management of the mappings towards the entity type Person. As it is possible to see, the mappings towards the type are dissected between equivalent classes and sub-classes. The mappings will be published as part of the identification ontology, that will then become an Open Linked

[1] http://code.google.com/p/google-refine/

Person	
Equivalent	SubClass
dbpedia:Person	umbel:MusicPerformer
schema:Person	dbpedia:Artist
foaf:Person	yago:LivingPeople
dul:Person	freebase:author
freebase:person	dbpedia:Politician
freebase:human	wn:synset-musician-noun-1
okkam:Person	schema:deceased_person
umbel:Person	dbpedia:Athlete
wn:synset-person-noun-1	umbel:BaseballPlayer
	umbel:Journalist
	wn:synset-celebrity-noun-1
	yago:Writer110794014
	dbpedia:Actor
	freebase:golfer
	...

Table 7.1: Entity Types Mapping Samples for Person

Vocabulary. For the moment, the mapping towards the classes and the attributes defined in the identification ontology are managed in a textual field, and every line of the text area represent a single mapping. We already planed future evolution of this application to become user friendly, and gradually evolve in an application suitable to be adopted in a community context.

7.1.1 Examples of Entity Type Contextual Mappings

It is important to remember that mappings proposed in this context are not meant to be static and absolutely correct from an ontological perspective, but rather they are conceived as contextual parameters of a process aimed at solving the entity matching problem. In this context, mappings do not aim only at capturing equivalent properties or classes, but rather aim at capturing also properties that can be considered subclasses or sub-properties of the features defined in the identification ontology. For this reason, we decided to rely on purely manual supervised approach in the definition of the mappings. This approach is time-expensive and does not necessarily scale, but allowed us to define a first set of mappings to evaluate the suitability of the knowledge-based solution. We plan to exploit the initial effort for the definition of the mappings to rely on some semi-automatic method supporting the mapping process. For example, [5], the authors propose a probabilistic method to guess the type of entity given the attributes. At the moment we defined 22 mapping for equivalent class, and 205 mapping for subclasses of the entity type Person. A sample of these mappings are presented in table 7.1. We also defined 22 mappings for equivalent classes, and 2322 mappings

Location	
Equivalent	SubClass
schema:Place	freebase:venue
dbpedia:Place	umbel:PopulatedPlace
yago:YagoGeoEntity	yago:GeoClassBridge
wn:synset-location-noun-1	schema:Park
okkam:location	SubClass
umbel:Place	dbpedia:city
geonames:Feature	wn:synset-city-noun-3
freebase:location	umbel:Village
dul:Place	yago:BridgesInNewMexico
opengis:_Feature	umbel:Town
	dbpedia:Station
	yago:WineRegionsInFrance
	schema:City
	wn:synset-lake-noun-1
	...

Table 7.2: Entity Types Mapping Samples for Location

for sub classes of the type Location. A sample of these mappings is presented in table 7.2. Finally, we defined 20 mappings for equivalent classes and 2468 mappings for subclasses of the type Organization. A sample of these mappings is presented in table 7.3. A more extensive list of mappings is presented in appendix A.

7.2 Semantic Harmonization of Features

In order to solve the problem of semantic heterogeneity affecting descriptions on the Semantic web we rely on the existence of a list of contextual mappings to align the name of features used in the descriptions with the canonical one defined in the Identification Ontology described in section 5. Also in this case, we are aware of the fact that some attributes might not present any type, or that a mapping for all the attribute types might not be available. However, it is important to notice that the proliferation in the creation of ontologies seems to be beginning to converge thanks to the "terms reuse policy" defined in the tutorial about *how to publish Linked Data on the Web* [20]. Hence, in this context the cost for the definition of mappings between ontologies is relevant but does not seem to be an unsolvable issue. Also in this case, we relied on the *SemanticMap* web application to defined and maintain the mappings of attributes towards the identification ontology features. Nevertheless, in the future we aim at defining automatic or semi-automatic solutions to guess the type of an attribute given its value, as proposed in [6] and in [144] among others. Furthermore, when the descriptions are *over-specified* and present a long list of attributes irrelevant for matching decisions, distance-based matching algorithms that do not consider the semantic

Organization	
Equivalent	SubClass
schema:Organization	umbel:Business
okkam:Organization	dbpedia:Company
dbpedia:Organisation	wn:synset-company-noun-1
umbel:Organization	schema:MusicalGroup
freebase:organization	umbel:NonProfitOrganization
wn:synset-organization-noun-1	yago:IslamicOrganization
foaf:Organization	wn:synset-institution-noun-1
foaf:Group	umbel:EducationalOrganization
dul:Organization	freebase:hokey_team
	freebase:employer
	yago:GirlGroup
	yago:1970sMusicGroups
	umbel:Club_Organization
	freebase:family
	...

Table 7.3: Entity Types Mapping Samples for Organization

of attributes can be heavily affected. This problem is partially solved by selecting the interesting attributes through the mappings towards the identification ontology, and then discarding the attributes that are not recognized to be relevant for matching.

Once the type is disambiguated, it is necessary to harmonize also the semantic of the attributes in the compared descriptions. Similarly to what was done for the entity types, also in this case we do not propose any automatic solution to discover the mapping, but we rely on a list of mappings manually defined. This choice is driven my the fact that the semantic harmonization it is not interpreted as traditional ontological matching, but rather it is interpreted as the result of the application of a set of contextual bridge rules implying some sort of generalization, besides ontological equivalence. The generalization of the mapping encompasses the concept of sub-property, but its complete formalization is out of the scope of this work, and thus we limit our analysis interpreting mapping as contextual bridge rules as described in section 5.6, supported by the theoretical framework defined in [29]. A sample of the mappings produced for the features associated with the type Person is presented in table 7.4.

7.2.1 Mappings for Features of Person

Table 7.4: Mapping for features of type Person

Feature	**Mappings URL**
affiliation	http://schema.org/affiliation, http://dbpedia.org/property/workplace, http://www.loc.gov/mads/rdf/v1#hasAffiliation, http://dbpedia.org/ontology/team, http://www.freebase.com/schema/baseball/baseball_coach/current_team_coaching, http://purl.org/net/nknouf/ns/bibtex#hasAffiliation, http://www.semanticdesktop.org/ontologies/2007/03/22/nco#hasAffiliation, http://dbpedia.org/property/coachingteams, http://dbpedia.org/ontology/party, http://www.ontotext.com/proton/protonext#ofParty, http://d-nb.info/standards/elementset/agrelon.owl#hasEmployer, ...
author	http://rdvocab.info/roles/author, http://www.freebase.com/schema/film/writer/film, http://purl.org/dc/elements/1.1/author, http://dbpedia.org/property/author, http://www.freebase.com/schema/music/lyricist/lyrics_written, http://dbpedia.org/property/novels, http://www.ontologydesignpatterns.org/ont/dul/IOLite.owl#isAuthorOf, http://www.ontotext.com/proton/protonext#authorOf, http://purl.org/ontology/mo/composer, ...
award	http://dbpedia.org/property/awards, http://schema.org/awards, http://www.freebase.com/schema/award/award_honor/award_winner, http://vivoweb.org/ontology/core#awardOrHonor, http://www.ontotext.com/proton/protonext#Award, http://rdvocab.info/Elements/award, http://dbpedia.org/property/academyawards, ...
birthdate	http://dbpedia.org/ontology/birthDate, http://dbpedia.org/property/dateofbirth, http://www.freebase.com/schema/people/person/date_of_birth, http://www.semanticdesktop.org/ontologies/2007/03/22/nco#birthDate, http://schema.org/birthDate, http://xmlns.com/foaf/0.1/birthday, http://www.w3.org/2006/vcard/ns#bday, http://www.ontotext.com/proton/protonext#birthDate, http://www.kanzaki.com/ns/whois#born, http://rdvocab.info/ElementsGr2/dateOfBirth, http://d-nb.info/standards/elementset/gnd#dateOfBirth, http://data.archiveshub.ac.uk/def/dateBirth, http://data.press.net/ontology/stuff/dateOfBirth, ...
birthmonth	http://dbpedia.org/property/monthofbirth, ...

Mapping for Features of Person, Continued on next page

birthplace	`http://dbpedia.org/property/birthPlace`, `http://rdvocab.info/ElementsGr2/placeOfBirth`, `http://www.freebase.com/schema/people/person/place_of_birth`, `http://www.freebase.com/schema/music/artist/origin`, `http://www.ontotext.com/proton/protonext#birthPlace`, `http://d-nb.info/standards/elementset/gnd#placeOfBirth`, `http://data.press.net/ontology/stuff/placeOfBirth`, ...
birthyear	`http://dbpedia.org/property/yob`, `http://dbpedia.org/ontology/birthYear`, `http://lod.taxonconcept.org/ontology/txn.owl#yearBorn`, ...
city of residence	`http://dbpedia.org/property/homeTown`, `http://dbpedia.org/property/residence`, `http://dbpedia.org/property/placeOfResidence`, `http://schema.org/homeLocation`, `http://rdvocab.info/ElementsGr2/placeOfResidence`, `http://schemas.talis.com/2005/address/schema#localityName`, ...
country of residence	`http://dbpedia.org/property/country`, `http://rdf.myexperiment.org/ontologies/base/country`, `http://www.w3.org/2006/vcard/ns#country-name`, `http://vivoweb.org/ontology/core#addressCountry`, `http://www.loc.gov/mads/rdf/v1#country`, `http://www.semanticdesktop.org/ontologies/2007/03/22/nco#country`, `http://lod.taxonconcept.org/ontology/txn.owl#country`, ...
date of death	`http://dbpedia.org/property/dateOfDeath`, `http://www.freebase.com/schema/people/deceased_person/date_of_death`, `http://dbpedia.org/property/died`, `http://purl.org/gen/0.1#death`, `http://data.archiveshub.ac.uk/def/dateDeath`, `http://www.ontotext.com/proton/protonext#deathDate`, `http://schema.org/deathDate`, `http://purl.org/vocab/bio/0.1/death`, `http://rdvocab.info/ElementsGr2/dateOfDeath`, `http://d-nb.info/standards/elementset/gnd#dateOfDeath`, `http://data.press.net/ontology/stuff/dateOfDeath`, ...
day of birth	`http://dbpedia.org/property/dayofbirth`, ...
deathplace	`http://dbpedia.org/property/deathplace`, `http://www.freebase.com/schema/people/deceased_person/place_of_death`, `http://dbpedia.org/property/cityofdeath`, `http://rdvocab.info/ElementsGr2/placeOfDeath`, `http://www.ontotext.com/proton/protonext#deathPlace`, `http://d-nb.info/standards/elementset/gnd#placeOfDeath`, ...

Mapping for Features of Person, Continued on next page

domain tag	`http://dbpedia.org/property/shortDescription`, `http://www.holygoat.co.uk/owl/redwood/0.1/tags/tag`, `http://schemas.talis.com/2005/dir/schema#tag`, `http://www.semanticdesktop.org/ontologies/2007/08/15/nao#hasTag`, `http://purl.org/dc/terms/subject`, `http://schema.org/keywords`, ...
email address	`http://xmlns.com/foaf/0.1/mbox`, `http://purl.org/b2bo#mailto`, `http://vivoweb.org/ontology/core#email`, `http://www.w3.org/2006/vcard/ns#email`, `http://www.w3.org/2000/10/swap/pim/contact#emailAddress`, `http://schema.org/email`, `http://swrc.ontoware.org/ontology#email`, `http://vivoweb.org/ontology/core#primaryEmail`, `http://www.aktors.org/ontology/portal#has-email-address`, ...
email address hashcode	`http://rdfs.org/sioc/ns#email_sha1`, `http://xmlns.com/foaf/0.1/mbox_sha1sum`, ...
end date	`http://www.freebase.com/schema/music/group_membership/end`, `http://dbpedia.org/ontology/activeYearsEndDate`, `http://www.freebase.com/schema/military/military_service/to_date`, `http://www.freebase.com/schema/education/education/end_date`, `http://spitfire-project.eu/ontology/ns/activityEnd`, `http://www.freebase.com/schema/business/employment_tenure/to`, `http://dbpedia.org/property/lastCupRace`, ...
fax	`http://sw-portal.deri.org/ontologies/swportal#hasFax`, `http://swrc.ontoware.org/ontology#fax`, `http://vivoweb.org/ontology/core#faxNumber`, `http://www.loc.gov/mads/rdf/v1#fax`, `http://schemas.talis.com/2005/address/schema#fax`, `http://schema.org/faxNumber`, ...
first name	`http://xmlns.com/foaf/0.1/firstName`, `http://rdfs.org/sioc/ns#first_name`, `http://www.ontotext.com/proton/protontop#firstName`, `http://purl.org/b2bo#firstName`, `http://schema.org/givenName`, `http://xmlns.com/foaf/0.1/givenName`, `http://www.w3.org/2006/vcard/ns#given-name`, ...
gender	`http://dbpedia.org/property/sexuality`, `http://www.freebase.com/schema/people/person/gender`, `http://dbpedia.org/ontology/gender`, `http://dbpedia.org/property/sex`, `http://xmlns.com/foaf/0.1/gender`, `http://www.semanticdesktop.org/ontologies/2007/03/22/nco#gender`, `http://rdvocab.info/ElementsGr2/gender`, `http://schema.org/gender`, ...

Mapping for Features of Person, Continued on next page

height	`http://dbpedia.org/ontology/Person/height`, `http://www.freebase.com/schema/people/person/height_meters`, ...
involved in	`http://www.freebase.com/schema/film/film_art_director/films_art_directed`, `http://www.freebase.com/schema/theater/theater_director/plays_directed`, `http://www.freebase.com/schema/music/producer/tracks_produced`, `http://purl.org/ontology/mo/produced_work`, `http://purl.org/vocab/frbr/core#producerOf`, ...
last name	`http://xmlns.com/foaf/0.1/surname`, `http://rdfs.org/sioc/ns#last_name`, `http://xmlns.com/foaf/0.1/family_name`, `http://www.ontotext.com/proton/protontop#lastName`, `http://purl.org/b2bo#lastName`, , `http://xmlns.com/foaf/0.1/lastName`, `http://schema.org/familyName` ...
member of	`http://dbpedia.org/property/movement`, `http://dbpedia.org/property/schoolTradition`, `http://dbpedia.org/ontology/religion`, `http://reference.data.gov.uk/def/parliament/memberOf`, `http://lexvo.org/ontology#memberOf`, `http://rdfs.org/sioc/ns#member_of`, `http://www.w3.org/ns/org#memberOf`, `http://reference.data.gov.uk/def/parliament/partyMemberOf`, `http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#isMemberOf`, `http://vivoweb.org/ontology/core#currentMemberOf`, `http://xmlns.com/foaf/0.1/member`, `http://schema.org/member`, ...
phone number	`http://xmlns.com/foaf/0.1/phone`, `http://www.w3.org/2006/vcard/ns#tel`, `http://vivoweb.org/ontology/core#phoneNumber`, `http://www.w3.org/2000/10/swap/pim/contact#Phone`, `http://www.loc.gov/mads/rdf/v1#phone`, `http://schema.org/telephone`, `http://www.ontotext.com/proton/protonext#hasMobilePhone`, ...
picture url	`http://xmlns.com/foaf/0.1/depiction`, `http://dbpedia.org/property/image`, `http://xmlns.com/foaf/0.1/img`, `http://lod.taxonconcept.org/ontology/txn.owl#individualhasImage`, `http://schema.org/image`, `http://xmlns.com/foaf/0.1/thumbnail http://data.press.net/ontology/stuff/hasImage`, ...

Mapping for Features of Person, Continued on next page

postal code	`http://schemas.talis.com/2005/address/schema#postalCode`, `http://schema.org/postalCode`, `http://data.lirmm.fr/ontologies/passim#postalCode`, `http://ogp.me/ns#postal-code`, `http://www.w3.org/2006/vcard/ns#postal-code`, `http://sw-portal.deri.org/ontologies/swportal#hasZipcode`, `http://www.ontotext.com/proton/protonext#ZipCode` ...
public istitutional id	`http://purl.org/goodrelations/v1#taxID`, `http://schema.org/vatID`, `http://purl.org/b2bo#nip`, `http://purl.org/goodrelations/v1#vatID`, `http://purl.org/spar/datacite/social-security-number`, `http://schema.org/taxID`, `http://www.semanticdesktop.org/ontologies/2007/01/19/nie#identifier`, ...
start date	`http://dbpedia.org/property/termStart`, `http://www.freebase.com/schema/music/group_membership/start`, `http://dbpedia.org/property/stateDate`, `http://purl.org/goodrelations/v1#vatID`, `http://www.freebase.com/schema/tv/regular_tv_appearance/from`, `http://dbpedia.org/property/debutdate`, `http://www.oegov.org/core/owl/gc#startDate`, `http://ns.nature.com/terms/dateStart`, `http://spitfire-project.eu/ontology/ns/activityStart`, `http://schema.org/startDate` ...
street address	`http://rdvocab.info/ElementsGr2/addressOfThePerson`, `http://sw-portal.deri.org/ontologies/swportal#hasStreetAddress`, `http://schemas.talis.com/2005/address/schema#streetAddress`, `http://www.w3.org/2006/vcard/ns#street-address`, `http://www.semanticdesktop.org/ontologies/2007/03/22/nco#streetAddress`, `http://schema.org/streetAddress`, `http://ogp.me/ns#street-address`, `http://www.w3.org/2006/vcard/ns#adr`, ...
title	`http://xmlns.com/foaf/0.1/title`, `http://dbpedia.org/property/honorificPrefix`, `http://www.freebase.com/schema/royalty/noble_title_tenure/noble_title`, `http://data.archiveshub.ac.uk/def/title`, `http://www.w3.org/2006/vcard/ns#title`, `http://xmlns.com/foaf/0.1/title`, `http://schema.org/title`, `http://schema.org/jobTitle` , `http://www.ontotext.com/proton/protonext#hasTitle`, `http://rdvocab.info/ElementsGr2/titleOfThePerson`, `http://d-nb.info/standards/elementset/gnd#titleOfNobility` ...

Mapping for Features of Person, Continued on next page

website	http://dbpedia.org/property/homepage, http://xmlns.com/foaf/0.1/homepage, http://www.freebase.com/schema/internet/blogger/blog, http://d-nb.info/standards/elementset/gnd#homepage, http://purl.org/ontology/mo/homepage, http://xmlns.com/foaf/0.1/weblog, http://www.semanticdesktop.org/ontologies/2007/03/22/nco#blogUrl, http://lod.taxonconcept.org/ontology/sci_people.owl#hasBlog, http://purl.org/ontology/mo/onlinecommunity, http://www.w3.org/2000/10/swap/pim/contact#webPage, ...

7.2.2 Mappings for Features of Location

A sample of the mappings produced for the features associated with the type Location is presented in table 7.5.

Table 7.5: Mapping for features of type Location

Feature	Mappings URL
area	http://www.freebase.com/schema/location/location/area, http://dbpedia.org/property/area, http://dbpedia.org/ontology/PopulatedPlace/area, http://dbpedia.org/ontology/team, http://dbpedia.org/property/landarea, http://dbpedia.org/property/areaMetroKm, http://www.telegraphis.net/ontology/geography/geography#landArea, http://www.mindswap.org/2003/owl/geo/geoFeatures20040307.owl#sqkm, http://aims.fao.org/aos/geopolitical.owl#landArea, ...
city	http://dbpedia.org/ontology/city, http://www.freebase.com/schema/location/mailing_address/citytown, http://sw-portal.deri.org/ontologies/swportal#inCity, http://www.mindswap.org/2003/owl/geo/geoFeatures20040307.owl#city_name, http://www.loc.gov/mads/rdf/v1#city ...
contains	http://dbpedia.org/ontology/subregion, http://dbpedia.org/property/frazioni, http://dbpedia.org/property/largestcity, http://dbpedia.org/property/capital, http://www.freebase.com/schema/location/place_with_neighborhoods/neighborhoods, http://data.ordnancesurvey.co.uk/ontology/spatialrelations/contains, http://ontologies.smile.deri.ie/pdo#contains, http://www.essepuntato.it/2008/12/pattern#contains, http://geovocab.org/spatial#Pi, http://www.ontotext.com/proton/protonext#containsLocation ...

Mappings of Feature for Location, Continued on next page

coordinate geometry	http://geovocab.org/geometry#geometry, http://www.opengis.net/ont/geosparql#hasGeometry, http://www.opengis.net/ont/geosparql#defaultGeometry, http://www.w3.org/2003/01/geo/wgs84_pos#geometry, ...
country	http://www.geonames.org/ontology#inCountry, http://www.geonames.org/ontology#parentCountry, http://www.freebase.com/schema/location/postal_code/country, http://www.freebase.com/schema/location/administrative_division/country, http://www.semanticdesktop.org/ontologies/2007/03/22/nco#country, http://lod.taxonconcept.org/ontology/txn.owl#country, http://sw-portal.deri.org/ontologies/swportal#inCountry, http://www.loted.eu/ontology#CY, http://www.geonames.org/ontology#countryCode, http://schemas.talis.com/2005/address/schema#countryName, ...
description	http://dbpedia.org/ontology/abstract, http://vivoweb.org/ontology/core#description, http://rdfs.org/sioc/ns#description, http://vocab.data.gov/def/fea#description, http://schema.org/description, ...
elevation	http://dbpedia.org/property/elevation, http://dbpedia.org/property/height, http://www.w3.org/2003/01/geo/wgs84_pos#alt, http://vocab.data.gov/def/fea#description, http://www.freebase.com/schema/location/geocode/elevation, http://schema.org/elevation ...
first level administrative parent	http://www.geonames.org/ontology#parentADM1, http://dbpedia.org/ontology/state, http://www.loc.gov/mads/rdf/v1#state, http://reference.data.gov/def/govdata/stateCode ...
fourth level administrative parent	http://www.geonames.org/ontology#parentADM4, ...
geocoordinate	http://www.freebase.com/schema/location/location/geolocation, http://www.georss.org/georss/point, http://schema.org/geo, http://www.mindswap.org/2003/owl/geo/geoFeatures20040307.owl#xyCoordinates, http://d-nb.info/standards/elementset/gnd#coordinates, http://rdvocab.info/Elements/stringsOfCoordinatePairs, http://www.w3.org/2003/01/geo/wgs84_pos#lat_long, http://purl.org/ontology/places#latlong ...

Mappings of Feature for Location, Continued on next page

is contained by	http://dbpedia.org/property/subdivisionName, http://dbpedia.org/ontology/lieutenancyArea, http://dbpedia.org/property/region, http://www.freebase.com/schema/geography/island/island_group, http://www.freebase.com/schema/location, http://www.freebase.com/schema/location/location/containedby, http://schema.org/containedIn, http://sw-portal.deri.org/ontologies/swportal#inRegion, http://www.ontotext.com/proton/protontop#subRegionOf ...
latitude	http://www.w3.org/2003/01/geo/wgs84_pos#latitude, http://www.freebase.com/schema/location/geocode/latitude, http://ogp.me/ns#latitude, http://www.w3.org/2003/01/geo/wgs84_pos#lat, http://dbpedia.org/property/latitude, http://schema.org/latitude, http://www.ontotext.com/proton/protontop#latitude, http://www.w3.org/2003/12/exif/ns#gpsLatitude, http://www.w3.org/2006/vcard/ns#latitude ...
latitude degree	http://dbpedia.org/property/latDegrees, http://dbpedia.org/property/latDegrees, ...
latitude direction	http://dbpedia.org/property/latDirection, http://dbpedia.org/property/latNs, ...
latitude minute	http://dbpedia.org/property/latMinutes, http://dbpedia.org/property/latM, ...
latitude second	http://dbpedia.org/property/latitudeseconds, http://dbpedia.org/property/latS, ...
location name	http://www.geonames.org/ontology#alternateName, http://www.freebase.com/schema/location/country/iso3166_1_shortname, http://www.geonames.org/ontology#officialName, http://xmlns.com/foaf/0.1/name, http://models.okkam.org/ENS-core-vocabulary#location_name, http://www.ontotext.com/proton/protonext#locationName, http://rdvocab.info/ElementsGr3/preferredNameForThePlace, http://schema.org/name, http://www.geonames.org/ontology#name ...
location type	http://dbpedia.org/property/subdivisionType, http://dbpedia.org/ontology/type, http://www.geonames.org/ontology#featureClass, http://www.freebase.com/schema/geography/geographical_feature/category ...

Mappings of Feature for Location, Continued on next page

longitude	http://dbpedia.org/property/long, http://www.w3.org/2003/01/geo/wgs84_pos#longitude, http://www.freebase.com/schema/location/geocode/longitude, http://schema.org/longitude, http://www.ontotext.com/proton/protontop#longitude, http://www.w3.org/2003/12/exif/ns#gpsLongitude, http://www.w3.org/2006/vcard/ns#longitude, http://ogp.me/ns#longitude ...
longitude degree	http://dbpedia.org/property/longtitudedegrees, http://dbpedia.org/property/longD ...
longitude direction	http://dbpedia.org/property/longDirection, http://dbpedia.org/property/longew ...
longitude minute	http://dbpedia.org/property/longtitudeminutes, http://dbpedia.org/property/lonMin ...
longitude second	http://dbpedia.org/property/longSeconds, http://dbpedia.org/property/longS ...
picture url	http://schema.org/photo, http://dbpedia.org/property/hasPhotoCollection, http://dbpedia.org/property/imageMap, http://schema.org/map, http://www.geonames.org/ontology#locationMap ...
postal code	http://dbpedia.org/ontology/postalCode, http://www.freebase.com/schema/location/mailing_address/postal_code, http://www.geonames.org/ontology#postalCode, http://schema.org/postalCode, http://www.loc.gov/mads/rdf/v1#postcode, http://sw-portal.deri.org/ontologies/swportal#hasZipcode ...
second level administration parent	http://www.geonames.org/ontology#parentADM2 ...
street address	http://www.freebase.com/schema/architecture/structure/address, http://dbpedia.org/ontology/address, http://schema.org/address, http://purl.org/b2bo#address, http://ogp.me/ns#street-address, http://www.w3.org/2006/vcard/ns#street-address, http://sw-portal.deri.org/ontologies/swportal#hasStreetAddress ...
third level administrative parent	http://www.geonames.org/ontology#parentADM3, ...
timezone	http://dbpedia.org/property/timeZone, http://www.freebase.com/schema/location/location/time_zones, http://dbpedia.org/property/timezoneDst, http://vocab.org/transit/terms/timezone, http://www.w3.org/2006/vcard/ns#tz, http://www.aktors.org/ontology/support#in-timezone, http://www.w3.org/2006/timezone, http://www.ontotext.com/proton/protonext#TimeZone ...
website	http://xmlns.com/foaf/0.1/homepage, http://www.w3.org/2000/01/rdf-schema#seeAlso, http://www.geonames.org/ontology#wikipediaArticle, http://dbpedia.org/ontology/wikiPageExternalLink ...

7.2.3 Mappings for Features of Organization

A sample of the mappings produced for the features associated with the type Organization is presented in table 7.6.

Table 7.6: Mapping for features of type Organization

Feature	Mappings URL
activity sector	http://www.freebase.com/schema/business/business_operation/industry, http://dbpedia.org/ontology/industry, http://www.freebase.com/schema/sports/sports_team/sport, http://dbpedia.org/ontology/ideology, http://www.freebase.com/schema/education/education/specialization, http://dbpedia.org/property/fields, http://www.freebase.com/schema/music/artist/genre, http://reegle.info/schema#sector, http://kmm.lboro.ac.uk/ecos/1.0#sector http://www.ontotext.com/proton/protonext#industryOf, http://umbel.org/umbel#relatesToMarketIndustry, http://purl.org/ontology/sport/discipline ...
activity start year	http://dbpedia.org/ontology/activeYearsStartYear, http://dbpedia.org/property/startYear, http://purl.org/ontology/mo/activity_start, http://dbpedia.org/ontology/openingYear, http://www.freebase.com/schema/book/newspaper_circulation/date ...
associated with	http://dbpedia.org/property/partner, http://dbpedia.org/property/sisterNames, http://dbpedia.org/property/sisterCompany, http://www.freebase.com/schema/religion/religious_organization/associated_with, http://www.freebase.com/schema/business/company_brand_relationship/company, http://dbpedia.org/property/associatedSchools, http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#associatedWith, http://www.freebase.com/schema/music/artist/label ...
award	http://www.freebase.com/schema/award/award_nomination/award, http://www.freebase.com/schema/award/award_winner/awards_won, http://schema.org/awards, ...
city	http://dbpedia.org/ontology/locationCity, http://dbpedia.org/ontology/hometown, http://dbpedia.org/property/city, http://vivoweb.org/ontology/core#addressCity, http://www.mindswap.org/2003/owl/geo/geoFeatures20040307.owl#city_name, ...

Continued on next page

color	http://dbpedia.org/ontology/colourName, http://www.freebase.com/schema/business/brand/colors, http://www.freebase.com/schema/education/educational_institution/colors, http://www.freebase.com/schema/sports/sports_team/colors, http://dbpedia.org/ontology/officialSchoolColour, http://vocab.org/transit/terms/color, http://rdf.muninn-project.org/ontologies/appearances#htmlColor ...
controlled by	http://www.freebase.com/schema/business/asset_ownership/owner, http://www.freebase.com/schema/organization/organization/parent, http://www.freebase.com/schema/business/shareholder/holding, http://dbpedia.org/ontology/parentOrganisation, http://dbpedia.org/ontology/owningCompany, ...
controls	http://www.freebase.com/schema/business/brand/includes_brands, http://www.freebase.com/schema/organization/organization/companies_acquired, http://dbpedia.org/ontology/childOrganisation, http://www.freebase.com/schema/organization/organization_relationship/child, http://dbpedia.org/property/subsid...
country	http://dbpedia.org/property/country, http://dbpedia.org/property/nat, http://www.w3.org/2006/vcard/ns#country-name, http://www.loted.eu/ontology#CY, http://reference.data.gov/def/govdata/country ...
description	http://dbpedia.org/ontology/purpose, http://dbpedia.org/ontology/abstract, http://usefulinc.com/ns/doap#description, http://vivoweb.org/ontology/core#description, http://purl.org/vocab/aiiso/schema#description, http://vocab.data.gov/def/fea#description, http://schema.org/description ...
dissolution date	http://dbpedia.org/ontology/extinctionDate, http://www.freebase.com/schema/exhibitions/exhibition_run/closed_on, http://www.freebase.com/schema/music/artist/active_end, http://www.freebase.com/schema/business/defunct_company/ceased_operations, http://purl.org/ontology/mo/activity_end ...
domain tag	http://dbpedia.org/property/background, http://purl.org/dc/terms/subject, http://dbpedia.org/property/freeLabel, http://schemas.talis.com/2005/dir/schema#tag, http://commontag.org/ns#label ...

Continued on next page

email address	http://xmlns.com/foaf/0.1/mbox, http://purl.org/b2bo#mailto, http://vivoweb.org/ontology/core#email, http://www.w3.org/2006/vcard/ns#email, http://www.w3.org/2000/10/swap/pim/contact#emailAddress, http://schema.org/email, http://swrc.ontoware.org/ontology#email, http://vivoweb.org/ontology/core#primaryEmail, http://www.aktors.org/ontology/portal#has-email-address, ...
email address hashcode	http://rdfs.org/sioc/ns#email_sha1, http://xmlns.com/foaf/0.1/mbox_sha1sum, ...
end date	http://www.oegov.org/core/owl/gc#endDate, http://d-nb.info/standards/elementset/agrelon.owl#hasEndDate, http://spitfire-project.eu/ontology/ns/activityEnd, ...
fax	http://sw-portal.deri.org/ontologies/swportal#hasFax, http://swrc.ontoware.org/ontology#fax, http://vivoweb.org/ontology/core#faxNumber, http://www.loc.gov/mads/rdf/v1#fax, http://schemas.talis.com/2005/address/schema#fax, http://schema.org/faxNumber, http://www.w3.org/2001/vcard-rdf/3.0/fax ...
foundation date	http://dbpedia.org/property/fDate, http://dbpedia.org/ontology/foundingDate, http://www.freebase.com/schema/sports/sports_team/founded, http://www.freebase.com/schema/organization/organization/date_founded, http://dbpedia.org/property/foundedDate, http://dbpedia.org/ontology/anniversary, http://www.oegov.org/core/owl/gc#foundedOn, http://schema.org/foundingDate, http://rdvocab.info/ElementsGr2/dateOfEstablishment ...
founded by	http://dbpedia.org/ontology/foundedBy, http://schema.org/founders, http://www.freebase.com/schema/organization/organization/founders, http://dbpedia.org/property/foundingPersonName, http://schema.org/founder, http://d-nb.info/standards/elementset/gnd#founder, http://d-nb.info/standards/elementset/agrelon.owl#hasFounder ...
geocoordinate	http://www.georss.org/georss/point, http://www.freebase.com/schema/location/location/geolocation ...
has foundation place	http://www.freebase.com/schema/organization/organization/place_founded, http://dbpedia.org/ontology/foundationPlace, http://dbpedia.org/property/foundedPlace, http://dbpedia.org/property/ceo ...

Continued on next page

has key people	http://dbpedia.org/ontology/leader, http://dbpedia.org/property/principal, http://www.freebase.com/schema/organization/leadership/person, http://www.freebase.com/schema/ice_hockey/hockey_team/coach, http://dbpedia.org/ontology/coach, ...
has location	http://dbpedia.org/property/regionServed, http://www.freebase.com/schema/organization/organization/locations, http://www.freebase.com/schema/organization/organization/geographic_scope, http://www.freebase.com/schema/broadcast/producer/location, ...
has members	http://www.freebase.com/schema/government/governmental_body/members, http://www.freebase.com/schema/music/concert_performance/band_members, http://www.freebase.com/schema/government/political_party/politicians_in_this_party, http://www.freebase.com/schema/basketball/basketball_roster_position/player, http://www.freebase.com/schema/education/education/student ...
has parts	http://dbpedia.org/ontology/division, http://www.freebase.com/schema/education/educational_institution/subsidiary_or_constituent_schools, http://www.freebase.com/schema/education/university/departments ...
involved in	http://www.freebase.com/schema/film/film_festival_sponsor/festivals_sponsored, http://www.freebase.com/schema/film/film_festival_sponsorship/festival, http://www.freebase.com/schema/conferences/conference_sponsor/conferences ...
is part of	http://www.freebase.com/schema/baseball/baseball_team/division, http://dbpedia.org/property/district, http://dbpedia.org/property/league ...
jurisdiction	http://www.freebase.com/schema/government/governmental_body/jurisdiction, http://dbpedia.org/ontology/jurisdiction, http://dbpedia.org/property/league, http://www.agls.gov.au/agls/terms/jurisdiction, http://purl.org/dc/elements/1.1/coverage, http://www.freebase.com/schema/organization/organization/geographic_scope, ...
latitude	http://dbpedia.org/property/latitude, http://www.w3.org/2003/01/geo/wgs84_pos#lat, http://www.freebase.com/schema/location/geocode/latitude, http://www.w3.org/2006/vcard/ns#latitude, http://dbpedia.org/property/lat ...

Continued on next page

longitude	`http://www.w3.org/2003/12/exif/ns#gpsLongitude`, `http://www.w3.org/2006/vcard/ns#longitude`, `http://dbpedia.org/property/longitude`, `http://schema.org/longitude`, `http://www.freebase.com/schema/location/geocode/longitude` ...
name	`http://xmlns.com/foaf/0.1/givenName`, `http://dbpedia.org/property/nonProfitName`, `http://dbpedia.org/ontology/teamName`, `http://dbpedia.org/property/fullName`, `http://dbpedia.org/property/companyName`, `http://dbpedia.org/property/nativeName`, `http://schema.org/name`, `http://purl.org/b2bo#name`, `http://schema.org/legalName`, `http://purl.org/goodrelations/v1#legalName`, `http://www.ontotext.com/proton/protontop#doingBusinessAs` ...
offers	`http://dbpedia.org/property/product`, `http://www.freebase.com/schema/dining/restaurant/cuisine`, `http://www.freebase.com/schema/cvg/cvg_developer/games_developed`, `http://www.freebase.com/schema/broadcast/producer/produces`, `http://www.freebase.com/schema/automotive/company/make_s`, `http://www.freebase.com/schema/film/production_company/films`, `http://www.freebase.com/schema/music/artist/track`, `http://www.freebase.com/schema/music/artist/album`, `http://www.freebase.com/schema/opera/opera_company/operas_produced`, `http://schema.org/makesOffer`, `http://data.archiveshub.ac.uk/def/isPublisherOf`, `http://purl.org/spar/cito/isCreditedBy`, `http://purl.org/goodrelations/v1#offers` ...
organization type	`http://dbpedia.org/ontology/type`, `http://dbpedia.org/ontology/governmentType`, `http://dbpedia.org/property/companyType`, `http://www.freebase.com/schema/education/educational_institution/` `school_typeof_type` ...
participant in	`http://dbpedia.org/property/battles`, `http://www.freebase.com/schema/music/artist/concerts`, `http://dbpedia.org/ontology/season`, `http://www.freebase.com/schema/music/artist/concert_tours`, `http://dbpedia.org/property/event`, `http://schema.org/event` ...
phone number	`http://xmlns.com/foaf/0.1/phone`, `http://www.w3.org/2006/vcard/ns#tel`, `http://vivoweb.org/ontology/core#phoneNumber`, `http://www.w3.org/2000/10/swap/pim/contact#Phone`, `http://www.loc.gov/mads/rdf/v1#phone`, `http://schema.org/telephone`, `http://www.ontotext.com/proton/protonext#hasMobilePhone`, ...

Continued on next page

photo	http://xmlns.com/foaf/0.1/depiction, http://dbpedia.org/property/imageName, http://dbpedia.org/ontology/thumbnail, http://schema.org/image, http://schema.org/logo, http://xmlns.com/foaf/0.1/logo, http://www.semanticdesktop.org/ontologies/2007/03/22/nco#logo, http://data.press.net/ontology/stuff/hasImage ...
postal code	http://dbpedia.org/ontology/postalCode, http://dbpedia.org/property/zipcode, http://sw-portal.deri.org/ontologies/swportal#hasZipcode, http://www.freebase.com/schema/location/mailing_address/postal_code, http://www.w3.org/2001/vcard-rdf/3.0/postal-code, http://schema.org/postalCode ...
previous name	http://dbpedia.org/property/formerNames, http://ndl.go.jp/dcndl/terms/previousName, http://www.freebase.com/schema/organization/organization/previous_names, http://www.freebase.com/schema/location/mailing_address/postal_code, http://www.w3.org/2001/vcard-rdf/3.0/postal-code, http://schema.org/postalCode ...
public institutional id	http://schema.org/vatID, http://purl.org/goodrelations/v1#taxID, http://purl.org/b2bo#nip ...
slogan	http://dbpedia.org/property/currentMottos, http://www.freebase.com/schema/business/brand_slogan/slogan, http://dbpedia.org/property/companySlogan, http://www.freebase.com/schema/education/educational_institution/motto ...
street address	http://www.w3.org/2001/vcard-rdf/3.0/street-address, http://models.okkam.org/ENS-core-vocabulary#street_address, http://dbpedia.org/property/address, http://www.freebase.com/schema/business/business_location/address, http://www.freebase.com/schema/book/newspaper/headquarters ...
street address	http://www.w3.org/2001/vcard-rdf/3.0/street-address, http://models.okkam.org/ENS-core-vocabulary#street_address, http://dbpedia.org/property/address, http://www.freebase.com/schema/business/business_location/address, http://www.freebase.com/schema/book/newspaper/headquarters ...
website	http://dbpedia.org/property/web, http://dbpedia.org/ontology/wikiPageExternalLink, http://dbpedia.org/property/homepage, http://xmlns.com/foaf/0.1/homepage, http://data.lirmm.fr/ontologies/passim#webSite ...

7.2.4 Remarks on Contextual Mappings

The list of mapping produced and briefly displayed in tables 7.4, 7.5 and 7.6 are the result of the effort of the author of this work to bootstrap the knowledge based solution. It is clear that the effort required to maintain such an extensive knowledge base is not scalable on a single person,

but it seems suitable for a community effort. For this reason, we are working on developing a web platform aimed at collecting users contributions in the maintenance both of the ontology and the mappings. This platform will make available through REST APIs the mappings produced by the community to support semantic harmonization tasks in external, third party applications.

Building communities around these task may not be easy at first, and besides we plan to sustain the effort in the long term, alternative complementary solutions have to be explored. For example, when no mapping is available, it could be important to categorize it based on other types of probabilistic knowledge. In [6], the author proposes PropLit as a pragmatic probabilistic method to exploit inverted index to guess the type of an attribute given its value. The method was experimentally evaluated relying on the billion triple challenge dataset[2], relying on an intuitive adaptation of cosine similarity related to the goal of the problem. Essentially, a large set of RDF documents was indexed relying on a Apache Lucene[3] index, which allowed to estimate how many times a specific value was used to instantiate a property. This allowed then conversely to estimate the type of an attribute given its value. However, as pointed out by the author, this approach has the limit of predicting the type of attributes whose value was previously indexed. Alternative approaches rely on the syntactic representation of the attribute value to estimate a classification probability, see for example [144]. These techniques, based on probabilistic models such as Conditional Random Field are commonly used also in Natural Language Process to solve the problem of Named Entity Recognition for different types of entities, see for example [90]. One one side, this type of solution guarantees a wide applicability of prediction based on a limited training set. On the other side, relying purely on the syntactic representations of the attribute value to guess the type, these cannot completely capture the semantic of the attribute type. Namely, it is possible to recognize that an attribute is date, but not that it is the date of birth, or the foundation date of a company. However, a combination of these two types of solution might be explored as future solution to support semi-automatic solution of the semantic heterogeneity problem.

[2] http://km.aifb.kit.edu/projects/btc-2010/

[3] http://lucene.apache.org/

7.3 Structural Harmonization of Features

Once the attributes are harmonized, we can exploit also specific syntactic knowledge about the harmonized features to attempt reducing possible differences among the descriptions based on structural heterogeneity (e.g. aggregating *first name* and *last name* to produce a name attribute).

1	**name**: Tom Jobim **name**: Antônio Carlos Brasileiro de Almeida Jobim **name**: Jobim, Antonio Carlos **birthdate**: 1927-01-25
2	**firstName**: Antônio Carlos **surname**: Jobim **family name**: de Almeida Jobim **day of birth**: 25 **month of birth**: 1 **year of birth**: 1927

Table 7.7: Examples structurally heterogeneous descriptions for date and name representation

The problem of structural heterogeneity is not second to the one of semantic heterogeneity when dealing with knowledge-based entity matching. In fact, if the attributes are not represented at the same level of granularity they cannot be compared accurately even thou they present matching set of information. For example, there exists attributes that can be represented as a whole, or split into sub parts. Consider for example a name, dates, street address, and geo-coordinates. These types of attributes can be represented as whole, as in description 1 of table 7.7, or split into their parts as in description 2 of table 7.7. These descriptions contain comparable attributes, but the different level of granularity in the representation of the data makes them practically not comparable.

In order to ease this type of problems we defined a set of transformation functions aimed at composing and de-composing attributes when these are recognized to be of some type from a syntactical perspective. An approach based on transformation function was proposed in [112], where the author propose to rely on a set of transformation function to produce a transformation graph that is used to compute a similarity score between descriptions. In this context we do not aim at relying on transformation functions to compute a score, but rather we rely on these transformations to normalize data representation where possible. The implemented process relies on the metadata defined in the identification ontology for the different features. In particular, for attributes of the type *data*, an extensible set of patterns is defined to parse the data string using Java SimpleDateFormat[4], and isolate its components. Examples of patterns are listed below:

```
dd-MM-yyyy
yyyy-MM-dd
yyyy MM dd
MM-dd-yyyy
...
```

If a date object can be parsed by any of the patterns, then the single parts of the data string can be used to create instances of the features describing the sub parts. In

[4] http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html

1	**name**: Tom Jobim **name**: Antônio Carlos Brasileiro de Almeida Jobim **name**: Jobim, Antonio Carlos **first name**: Tom **first name**: Antonio Carlos **name**: Antonio Carlos Jobim **first name**: Antônio Carlos Brasileiro **family name**: Jobim **family name**: de Almeida Jobim **birthdate**: 1927-01-25 **day of birth**: 25 **month of birth**: 1 **year of birth**: 1927
2	**firstName**: Antônio Carlos **family name**: Jobim **name**: Antônio Carlos de Almeida Jobim **name**: Antônio Carlos Jobim **family name**: de Almeida Jobim **day of birth**: 25 **month of birth**: 1 **year of birth**: 1927 **birthdate**: 1927-01-25

Table 7.8: Examples structurally heterogeneous descriptions transformed to ease heterogeneity problem

the case of *date* feature, the features produced would be *day of birth*, *month of birth* and *year of birth*.

Similar approach was taken with names of person and geo-coordinate. In order to decompose a name string, we relied on Yago NAGA Javatools implementation of name utilities[5]. Therefore, after applying such transformation functions, the descriptions outlined in table 7.7, become like the one outlined in table 7.8. It is important to remember that besides decomposition patterns, it is possible to apply also composition patterns for names. In fact, given the parts of names, it is possible to compose few syntactical variations of the name attribute to ease syntactical matching issues.

For example, a description about an entity could present all necessary information to take matching decision in a textual paragraph, whereas another one could present the same amount of information shredded into a list of attributes. Take for example the descriptions presented in table 7.9. For a person, comparing these descriptions is quite natural. In fact, many essential information about the Brazilian musician Tom Jobim are present both in description 1 and description 2. However, details such as date of birth and date of death, the complete name, and the city of death are present only in the descriptive paragraph of description 2. Thereby, considering the semantic of the attributes, only few attributes can be compared (i.e. name, domain tag, and occupation).

An effective way to ease this type of structural heterogeneity would be to extract features from textual paragraph so that they can be compared with the other features. For the moment, we don't tackle particularly this problem.

[5] `http://www.mpi-inf.mpg.de/yago-naga/javatools/`

1	**affiliation**: MCA Records **website**: `http://www.tomjobim.com.br/` **domain tag** Category:Msica Popular Brasileira pianists **occupation**: Singer **name**: Tom Jobim **domain tag**: Category:Brazilian singer-songwriters **birthdate**: 1927-01-25 **affiliation**: Philips Records **domain tag**: Category:Verve Records artists **birthplace**: Rio de Janeiro, Brazil **name**: Antônio Carlos Brasileiro de Almeida Jobim **name**: Jobim, Antonio Carlos **domain tag**: Msica Popular Brasileira **last name**: Jobim **occupation**: solo singer **website**: `http://www.thebraziliansound.com/jobim.htm` **domain tag**: Category:Cardiovascular disease deaths in New York **city of residence**: Rio de Janeiro (state) **affiliation**: A&M Records **date of death**: 1994-12-08 **domain tag**: Category:Grammy Award winners **affiliation**: Decca Records
2	**name**: Antônio Carlos Jobim **picture URL**: `http://userserve-ak.last.fm/serve/_/2245888/Antnio+Carlos+Jobim.jpg` **domain tag**: bossa nova **domain tag**: jazz **domain tag**: brazilian **domain tag**: mpb **domain tag**: latin **description**: Antônio Carlos Brasileiro de Almeida Jobim (born January 25, 1927 in Rio de Janeiro, Brazil December 8, 1994 in New York City), also called Tom Jobim, was a Brazilian composer, arranger, singer, pianist and perhaps the greatest legend of bossa nova. Jobim's compositions, many performed by Joao Gilberto, gave birth to the genre in the early 1960s. Jobim's roots were planted firmly in the works of Pixinguinha, a legendary musician and composer who, in the 1930s, began the development of modern Brazilian music. He was also influenced by the music of French composer Claude Debussy and by jazz. **occupation**: Musician **occupation**: Artist

Table 7.9: Examples structurally heterogeneous matching descriptions

Chapter 8

Building Rules for Open Entity Matching

The construction of rules for open entity matching is a central point of the solution proposed in this work. In chapter 4, we outlined the ingredients necessary to define a knowledge based solution for entity matching. In particular, in section 6 we formally defined the entity matching rules, and several tools for their interpretation and application and satisfaction. We also claimed to frame the rule based solution under the Open World Assumption, defining precise semantic of the rules, and formal interpretation related to their satisfaction or partial satisfaction. The core of the solution is that when no rule can be completely satisfied, the classification result is unknown.

In this section we describe two complementary and opposite approaches we relied on to construct entity matching rules. On the one hand we adopt a top down approach, aimed the exploiting the results of ontological analysis about meta-properties of the features defined in the identification ontology. On the other hand, we want to rely on a bottom-up approach aimed at extracting entity matching rules starting from a set of labeled samples relying on machine learning techniques.

Relying on sole top-town matching rules would hardly produce satisfying matching results, as considering single attributes we are very unlikely to produce many positive matching decisions. In fact, inverse-functional attribute suitable to be employed in an open context are rare and rarely usable. However, it would be a pity to not match descriptions when we can easily draw conclusion of such types of attributes. On the other side, top-down rules are very likely to exploit deduction related to functional properties, that can be used to produce negative matching decisions. The top-down construction of rules is described in section 8.1.

Learning very general matching rules starting from a training set of labeled samples seems to be extremely challenging. In fact, for how heterogeneous and complete a training set can be, it can never be representative of the whole possible varieties of representations of the entity types considered. Nevertheless, being aware that a complete solution is practically unachievable, we believe that modern machine learning techniques can help in capturing part of the knowledge employed by people in solving this task and produce reliable entity matching rules. For this reason, in this work we implemented experiments aimed at extracting matching rules in a bottom up fashion, testing different hypothesis and learning approaches. A detailed description of the process leading to the bottom up construction of entity matching rules is presented in section 8.2.

Both top-down and bottom up approach present shortcomings related to the fact that both present limits in the dealing with the entity matching problem in the open world. Integrating the rules result of these complementary process can lead to a more complete and sound set of matching rules. However, the integration of rules coming from different sources creates several issues we have to deal with. A detailed discussion about the top-down bottom up rules integration process is presented in section8.3.

8.1 Top-down Rule Construction

In section 8.2, we present a method for eliciting entity matching rules in a bottom up fashion, relying on machine learning techniques. This approach is essential to estimate numerical threshold on the types of attributes, and their weight when considering a score-based similarity metrics. However, it is clear that this approach has drawbacks. One one side, we can attempt to learn what are the combination of attributes that people would use to take matching decision, but on the the other side we introduce a bias on what can be learned given by the richness of the sources chosen to form the training set. A simple example of this bias is given by the fact that attributes that are considered to be excellent identifiers such as email address and social security number are hardly contained in public sources. One could argue that this bias can be reduced by extending the set of considered sources to cover all possible cases. However, pursuing this type of completeness seems to be hardly achievable and does not seem convenient. First of all, to establish when all the cases are covered becomes fuzzy, as one would require to know about all the existing data sources. Complete knowledge, and thus optimal solution, is not achievable when dealing with the Web and its complexity. However, an incremental approach where further sources are added with time seems to be a viable path to improve the capability of learning entity matching rules.

For this reason we aim at integrating bottom-up extracted rules with a set of rules defined relying on ontological analysis principles. In particular, we are going to rely on the meta-properties for identification outlined in section 5.4 to produce a set of positive and negative matching rules aimed at extending and increasing the matching capability of the knowledge-based solution proposed so far. It is important to underline the fact that the method we are going to propose relies on pure ontological analysis methods, and thus it can be applied to extract matching rules for any type of entity depending on the ontology of reference. The meta-properties presented in section 5.4 are defined extending the set traditional ontological (functional and inverse-functional) meta-properties proposing an analysis based on main dimension:

- *Time dimension.* It is important to keep into consideration how and whether values of certain properties can evolve in time. In particular, we propose to dissect properties between synchronic and diachronic. Synchronic properties are properties useful for identification at a fixed point in time. Diachronic properties are useful for identification at different points in time.

- *Scope dimension.* It also important to keep into consideration the scope of interpretation of properties. In particular, we propose to dissect properties between global and private. Global properties can be interpreted uniformly in any context. Private properties can be interpreted correctly only in a limited bounded context.

It is important to notice that stretching the concepts proposed above from a philosophical perspective is out of the scope of this work, and we pragmatically limit our analysis to what is useful in supporting the solution of the entity matching problem in an open context. It is also important to notice that any top down rule defined in this context is assumed to be valid neglecting the problems related to imperfect string similarity metrics, or odd matching cases. In fact the impact of such rules will have to be evaluated through experiments.

Functional Diachronic properties with a global scope can be used to perform the following reasoning: if p is functional and diachronic, and there are two descriptions $d_1 \in D$ and $d_2 \in D$, if $\exists p((a_1 \in \delta_d(d_1) \wedge p \in a_1) \wedge (a_2 \delta_d(d_2) \wedge p \in a_2))$ then if $v_{a_1} \neq v_{a_2}$ implies d_1 and d_2 do not match. Intuitively, if there can be only one value at any time for a specific property of an entity, if the value of that property does not match in two descriptions, these cannot be about the same real world entity. Thereby, it is possible to construct a negative matching rule $\rho :< p, <, 1.0, NON\_MATCH >$. These types of properties are for example birthday, birthplace for a person, longitude and latitude for a location, foundation date and foundation place for a company, among others.

Conversely, *Inverse-functional Diachronic properties with a global scope* can be used to perform the following reasoning: if p is inverse-functional and diachronic, and there are two descriptions $d_1 \in D$ and $d_2 \in D$, if $\exists p((a_1 \in \delta_d(d_1) \wedge p \in a_1) \wedge (a_2 \delta_d(d_2) \wedge p \in a_2))$ then if $v_{a_1} = v_{a_2}$ implies d_1 and d_2 match. Intuitively, if the value of a property can be associated only to one entity, then if two descriptions present the same value for that property, the it is possible to conclude that the descriptions are about the same real world entity. Thereby, it is possible to construct a positive matching rule $\rho :< p, >=, 1.0, MATCH >$. These types of properties are for example SSN, personal email, geocoordinate for locations, VAT number for organization among others.

In the previous paragraphs we defined principles for the definition of rules relying on ontological properties. However, functional and inverse-functional diachronic properties with a global scope are quite rare. On the other side, we can easily think about functional properties that are global but not strictly diachronic as identity criteria. Examples of these properties are surname, wife, city, state and street address of residence among others. These types of properties being not strictly diachronic do not guarantee stability in time. Thereby, when interpreted in the open asynchronous global space of the Web, they cannot be used reliably to produce negative matching rules by themselves. A similar analysis can be done for inverse-functional not rigidly diachronic properties.

However, the fact that a property is not strictly diachronic does not imply that it is also strictly synchronic. Namely, its value does not necessarily changes in time, it may change but not necessarily. Adopting the notation of *OntoClean*, one could say that these are semi-diachronic. We could exploit the soft or semi-diachronicity of both functional and inverse-functional properties to collectively produce some matching rule. The intuition is that if the single property alone is not suitable to produce a rule, a combination of them is likely to produce a robust rule. Hence, in this work we propose to explore functional and inverse-functional non-striclty diachronic properties for guessing positive and negative matching rules. In particular, we propose:

- given an enumeration of m functional and inverse-functional properties, produce all permutations of cardinality k, and interpret each permutation as a positive matching rule.
- given an enumeration of m functional properties, produce all permutations of cardinality j, and interpret each permutation as a negative matching rule.

As a further constraint, we assume that all permutations must contain the attribute name, as a basic minimal conditions shared by all the guessed rules. The optimal size of k and j has to be estimated through experiments. Once we generated these rules based on ontological analysis over properties defined in the Identification Ontology, we have to combine them with the rules result of the bottom up approach. Intuitively, this process should be different from the merging process proposed for rules learned using different classifiers. The conceived solutions are presented in section 8.3. The impact of the top-down generated rules will be evaluated through experiments in chapter 10.

8.2 Learning Rules for Open Entity Matching

In section 8.1 we present a method for building entity matching rules relying on meta-properties of the features defined in the Identification Ontology. Such rules can contribute in taking matching decisions, but they are likely to be applied on relatively small set of cases. In fact, most of the top-down positive matching rules rely on sort of global identifiers (e.g. email address, and public institutional id), which we can hardly find in web resources.

In this context we want to describe a method for learning entity matching rules suitable for the *Open World Entity Matching* as a reformulation of the traditional entity matching problem in the context of the Web, assuming its scale, mutability, heterogeneity and possible inconsistencies. As mentioned in chapter 2, there are several issues we have to deal with:

1. **semantic heterogeneity**: descriptions are often represented according to different vocabularies and schemas.
2. **structural heterogeneity**: attributes are often represent at different level of granularity. There are descriptions that wrap most of the information in a generic descriptive paragraph, and others that rely on a wide set of attributes.
3. **underspecification**: a description could be underspecified with respect to its interpretation in an open context, possibly omitting implicit contextual information, and thus causing problems of ambiguity (e.g. a description of a restaurant could omit the name of the city among the attributes used for the description, using only on the street name and number).
4. **over-specification**: a description could be over-specified, presenting an excessive amount of information that is relevant or interpretable only within a specific context [111].

Intuitively, the larger the context that is considered when making a matching decision, the higher the possibility of dealing with possibly underspecified descriptions, and learned rules should be able to capture this aspect and thus we should avoid learning rules tuned on underspecified samples. Bottom up rules extraction should make no assumption about the quality of the information involved in a matching process, and the rules should consider the Open World Assumption[1], considering the *unknown* matching cases as an explicit possibility as described in section 6.

The process of learning matching rules was already studied in different contexts both information systems as shown in section 3.1.3 and (semantic) web context in section 3.2. Defining rule based system to solve this type of problem is attractive, as usually rules are self-explanatory, explicit, and also manually configurable without any specific know-how about the way they will be employed. Consider for example SILK [147], the most popular solution for entity matching on the Linked Data. The solution basically consists in manually defining and maintaining matching rules to match descriptions in pairs of datasets. However, the main drawback of manually defined rules, is that they require a large amount of human effort to define an maintain them [58]. In the past years, many works proposed solution to learn rules for entity matching solution, among others, consider [61, 80, 143, 161, 149, 88, 41, 118]. However, all of them proposed and evaluated optimized methods for solving entity matching in pairs of dataset. Such rules proved to be effective on the evaluated datasets, but their application in an open generic context was never considered. In the following sections we propose a method for learning matching rules that are not necessarily specific for any pairs of dataset and that in principle can be employed to match any pairs of description. Furthermore, our goal is to capture in the rules the common knowledge used by people in solving the entity matching problem under the assumption that, in an open context affected by knowledge deficiency, no automatic method can top human supervised matching decision.

The process for the extraction of entity matching rules consists in a sequence of simple steps:

- Collect samples of descriptions representative of the heterogeneity of the Web;
- Harmonize the semantic of the attributes;
- Label pairs of samples to create a training set;
- Extract Rules relying on Decision Tree learner;

These steps are depicted in figure 8.1, and described in the following sections. In particular, section 8.2 presents a description of the training set generation process, outlining details about the data samples collection and the process of the selection of sample pairs for labeling. Section 8.2.4 presents some details about the method used to learn a decision tree, including some consideration about the usage of sample filters.

[1] The truth-value of a matching statement is independent of whether or not it is *known* to be true by any single observer.

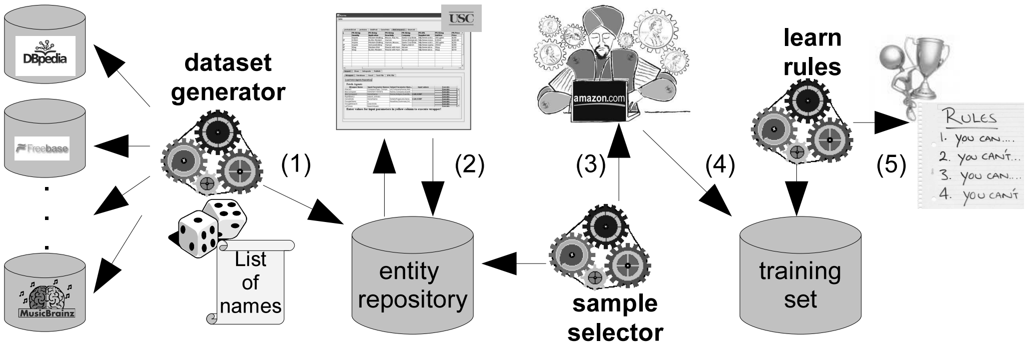


Figure 8.1: Rules Extraction Process

8.2.1 Data Samples Collection

In this section we formalize a process to gather entities' descriptions that are going to form the dataset on which matching rules have to be learned. Aiming at a knowledge based solution, we need to specify *a priori* what are the types of entities we intend to rely on. At this point, we assume that a set of types $T : \{t_1, ..., t_n\}$ have been specified in the Identification Ontology as described in section 5. As mentioned in the introduction of the chapter, we do not aim at learning rules for pairs of dataset, but we rather aim at learning more generally applicable rules for each of the types $t_i \in T$.

In order to collect data samples, we need a set data sources $S : \{s_1, ..., s_m\}$ we can use to gather description samples for our purposes for each of the types $t \in T$ defined in the ontology.

In particular, we define a data source $s_i \in S$ is of the tuple:$s_i :< u_i, l_i, r_i >$, with u_i representing the URL pointer identifying the source, l_i the URL pointer to the search/lookup service of the source and r the pointer to the resolver service allowing to get the complete description of an entity given its (local) identifier. Notice that for Cool Uri's [132], the pointer to the resources allows also the access to the complete description. However, not all the sources rely on Cool URIs to identify resources, and thus access to description is mediated by some sort of resolver. Lets define D_s as the set descriptions available through a source $s \in S$. Lets define $d_s \in D_D$ as the sample of information available in the data source S as it is.

Each of the data sources has to associated with a wrapper W_s. A wrapper $W_s : D_S \times C \rightarrow D$ is a function that transforms a description $d_s \in D_S$, into a description $\mathcal{D} = \left\{a_1^{[\mathcal{M}]}, ..., a_n^{[\mathcal{M}]}\right\}^{[C]}$ and $\mathcal{D}_2 = \left\{b_1^{[\mathcal{M}]}, ..., b_s^{[\mathcal{M}]}\right\}^{[C]}$, where a_i and b_i are attributes of the form (α_i, v_i) with α as possibly empty attribute name and v as attribute value, and C is a set of contextual information attributes to be appended in the transformed description.

Given a set of data sources S, and a set of wrappers W, we now need to define a process to gather sample descriptions from the data sources. Cognitive science studies proved that names are the first type of attribute used to search the main entity types [6]. For this reason, we assume the existence a list of *names* for the type of entity considered. The process of sample collection consists then in randomly selecting a name from the list available, and submit it to all the data sources lookup services to randomly gather samples. To reduce as much as possible the bias related to the samples data collection, the list must be extensive.

More formally, we define $Q_s : S \times N \to D_s$ is a query function that, given a source and name, relying on the search service pointer l_s and a resolver service pointer r_s of a source $s \in S$, retrieves a set of description $\{d_s \in D_s\}$. The dataset is generated relying on the following process:

```
n = getRandomName(List names);
getEntity(n){
    for each source s in S {
        List idList = s.search(Qs(n));
        for each id in idList {
            Ds raw = s.get(Qs(id));
            D e = Ws.wrap(raw);
            e = e.appendContext(Cs);
            store(e,s.u,n);
        }
    }
}
```

Namely, given a random name n, for each of the source a query seeded with n a list of identifiers is retrieved, and then those identifiers are used to retrieve raw descriptions, which must be wrapped in a common format and stored. It is important to notice that this approach is comparable to a sort of rough attribute-based blocking system capable of collecting from distributed and heterogeneous sources samples that may possibly refer to the same real world entity. This process has to be repeated for all the types considered in the Identification Ontology.

For specific types of entities, we can hardly assume an extensive list of events' names we can use to collect data samples. Thereby, we propose an alternative approach that can integrate the outlined one. That is, we propose to rely on some sort of iterative, name driven crawling process across sources. Intuitively, given a relatively small list of seed queries, we propose to lookup for descriptions samples on one source (the pivot source), and then iterating on the results, extract attributes of type 'name' to be used as lookup queries to all the other sources. The process is repeated in a fixed number of iterations, changing the pivot source.

```
getEntities(){
  n = getRandomName(List names);
  Source pivot;
  List idList = pivot.search(Qs(n));
  for each id in idList {
      Ds raw = pivot.get(Qs(id));
      D e = Wpivot.wrap(raw);
      e = e.appendContext(Cs);
      store(e,pivot.u,n);
  }
  List names = getNames(idList);
  for each name n in names{
      for each source s in S\{pivot} {
          List idList = s.search(Qs(n));
          for each id in idList {
              Ds raw = s.get(Qs(id));
              D e = Ws.wrap(raw);
              e = e.appendContext(Cs);
              store(e,s.u,n);
          }
      }
}
}
```

This process allows to gather possibly matching descriptions among the data sources, but does not allow us to index the retrieved descriptions around the used seed query.

The dataset generator components includes description extractors for entities of type "person", "location" and "organization". For the entity type person we implemented extractors for data sources available online such as:

- DBPedia[2], as a generic source of information presenting descriptions for 416.000 persons, 526.000 places and 169.000 organization. DBPedia is a crowd-sourced community effort to extract structured information from Wikipedia and thus the information about persons available through DBPedia are going to be likely to about famous people, locations and organization.

- Freebase[3], as a generic source for information about 23 millions of entities including person from US Census, companies, wikipedia, and other sources, locations and organizations. Also in this case we are likely the descriptions are likely to be related to famous people, or at least people cited in some public source, locations and organization.

- Musicbrainz[4], is a structured open online database for music offering information about 660.000 musicians including people and groups.

- LastFM[5], is a music website, founded in the United Kingdom in 2002 that claimed 30 million active users in March 2009. LastFM offers 'scrobbler' APIs to gather information about 500.000 artists including people and groups.

[2] http://dbpedia.org/About

[3] http://www.freebase.com/

[4] http://musicbrainz.org/

[5] http://www.last.fm/

- Factual[6], is an aggregator and provider of open data, which it provides access to through web service APIs and reusable, customisable web applications. Factual contained data about US politicians, all the governors in the history of US, baseball athletes and a large set of health care providers. Furthermore, manages geographical information for a large set of places and organizations such as restaurants, hotels, wine producers and business organization in general.

- OpenCongress[7] a non profit, non-partisan public source of information about congress member in the US.

- Okkam[8] ENS, a system for the management of globally unique identifiers for entities on the web. Any identifier is associated with a profile (set of information) from different sources and manually maintained. Thus also the Okkam ENS is a generic source of profiles that can be exploited to perform matching experiments. The ENS manages over 6 millions of locations, 380.000 person and more than 100.000 organizations.

- Geonames[9], a geographical database covers all countries and contains over 8,251,333 placenames that are available for download free of charge.

- OpenCorporates[10], is an open database about the corporate world containing information about more than 51 million companies all around the world.

For each of the sources exits a lookup service, that given a keyword and the type of entity of interest, returns either a list of results (e.g. Okkam, Factual, LastFm, Geonames), or a list of pointers (or identifier) that allowed to retrieve documents containing information (e.g. DBPedia, Freebase, MusicBrainz, OpenCongress, OpenCorporates). Some of the sources required user id registration (MusicBrainz, LastFM, Factual, OpenCongress, OpenCorporates) the other did not require any registration. Every data source returned results in different format. Some returned XML documents (MusicBrainz, LastFM), other JSON documents (Okkam, Freebase, OpenCongress, Factual, OpenCorporate) and other RDF XML (DBPedia). The latest version of Factual and Geonames provided a driver library, supporting the retrieval of description without dealing directly with data representations formats. The Factual dataset is structured in tables, and thus the lookup us service requires the specification of the dataset to query for each lookup query.

[6] http://www.factual.com

[7] http://www.opencongress.org

[8] http://api.okkam.org

[9] http://www.geonames.org/

[10] http://opencorporates.com/

Source	#desc	#attr	avg. #attr	attr STD
dbpedia	2927	248117	84.77	41.49
factual	3293	75623	22.96	5.1
freebase	2031	155160	76.40	67.51
lastfm	1253	42476	33.90	16.73
musicbrainz	1726	14879	8.72	5.4
okkam	1382	16178	11.70	1.05
opencongress	821	11607	14.13	2.86
TOTAL	13433	449659	33.45	37.49

Table 8.1: Person Dataset Collected Description

Person Samples Collection

Some data selection process required multiple queries, to allow gathering information related to the person. For example, MusicBrainz and LastFM managed separately the primary information about the artist, from the albums and tracks released. Thus, for each artist multiple queries to the data sources are produced in order to collect a complete set of information.

The queries to the data sources listed above are composed by randomly selecting words from a list of more than 738.000 names, obtained by processing the English DBpedia "persondata" dataset[11] according to the process outlined in section 8.2.1. Given a random name n, for each of the sources s considered, a search query $Qs(n)$ is submitted aimed at retrieving a list of identifiers. For each of these identifiers, a raw description is then retrieved from the considered source, and wrapped as an entity description $\mathcal{D}$, extended with contextual information, and stored together with the source URL and the name query used. We collected 13433 entities description

[11] http://wiki.dbpedia.org/Downloads32#persondata

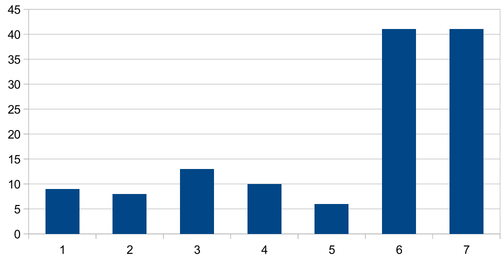


Figure 8.2: Number of queries per data sources returning samples

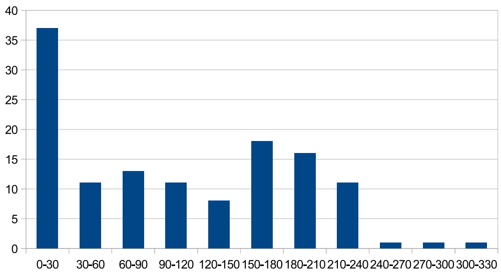


Figure 8.3: Response size distribution in blocks of samples retrieved

with 256 seed queries (e.g. Adam, Aspasius, Jenkins, Robbemond, Zhou, etc.). The description present a total number of 449659 attributes. The complete list of queries used for person is presented in appendix B.

In figure 8.2 we present the distribution of the number of queries based on number of sources responding for each of them. As shown in the picture, the largest part of the queries allowed us to retrieve samples from 6 or 7 sources, and only a small number of queries had response from few sources. This fact increases the possibility of gathering possibly overlapping descriptions and proves the feasibility of the seed query information retrieval approach. In figure 8.3 we present a view of the response size distribution for each of the queries. More than 30 queries queries allowed to retrieve at most 30 samples among all sources. This is probably due to the high level of selectivity of the query (e.g. Aspasius). On the other side, a very small number of queries were so general to retrieve more than 240 description among all the sources. As shown in the graph, the large majority of queries retrieved allowed to retrieve between 30 and 240 samples.

In table 8.1 we present a description of the data retrieved using this method. As it is possible to see the sources produced different results in terms of average number of attributes per description retrieved. Furthermore, some of the sources show how the average value is not reliably presenting a very high standard deviation. This aspect is quite representative of the structural heterogeneity and different of richness presented by different data sources. Collecting descriptions of person from different sources, with different scopes and target usage, we aim at supporting the learning of matching rules suitable to be employed as general mean of identification of person on the web. Nevertheless, we are aware the we are covering a relatively small set of possible representations of the entity type person. If experimental evaluation proves the process to be successful, we aim at pursuing completeness in this direction. This, among others, would be a good reason to promote and sustain the creation of a community supporting and sharing the effort of an increasingly extension of the possible representation of person covered by the set of rules.

The dataset generated is very sparse with more than 3021 different attributes types, and only a relatively small set of attribute types shared among descriptions. This is mostly due to the large variety of sometimes very specific properties used in sources such as DBpedia and Freebase. We believe that this semantic heterogeneity is representative of the one affecting entity matching on the web, and thus it is suitable to support the learning of rules that should be applied in this context. This problem mostly challenges the semantic harmonization task, which becomes particularly cumbersome, as shown in section 7.2.1. The number of harmonized attributes at the moment of

Source	#desc	#attr	avg. #attr	attr STD
dbpedia	2959	204008	68.95	45.88
factual	7791	173045	22.21	45.00
freebase	7001	406298	58.03	73.48
okkam	3377	42984	12.72	5.82
geonames	6742	115588	17.14	6.93
TOTAL	13433	449659	33.45	37.49

Table 8.2: Location Dataset Collected Description

writing is 317010 attributes, with 132649 attributes non harmonized. In average, every description presents 23.60 harmonized attributes, with a standard deviation of 27.12. The number of non harmonized attributes is in average 9.87, with a standard deviation of 14.04. The overall coverage ratio at the moment of writing is 0.70 for entity type person.

Location Samples Collection

The queries to the data sources listed above are composed by randomly selecting words from a list of more than 5.422.000 names, obtained by processing the Geonames RDF Dump dataset[12] according to the process outlined in section 8.2.1. Given a random name n, for each of the sources s considered, a search query $Qs(n)$ is submitted aimed at retrieving a list of identifiers. For each of these identifiers, a raw description is then retrieved from the considered source, and wrapped as an entity description $\mathcal{D}$, extended with contextual information, and stored together with the source URL and the name query used. We collected 27870 entities description with 838 seed queries (e.g. Aitken Cove, Aristovo, Ban Tham, Caleta, Zhou, etc.). The description present

[12] `http://download.geonames.org/all-geonames-rdf.zip`

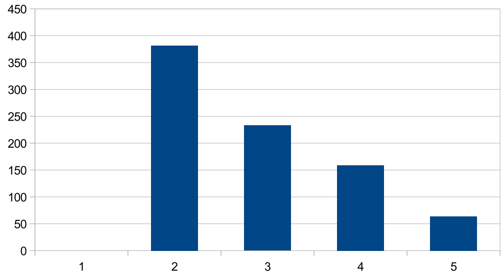


Figure 8.4: Number of queries per data sources returning samples

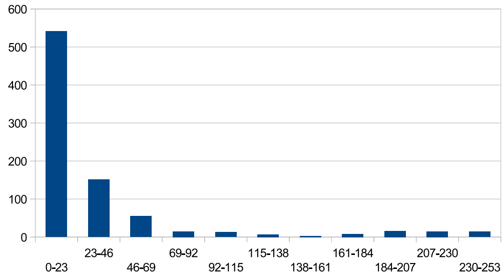


Figure 8.5: Response size distribution in blocks of samples retrieved

a total number of 941923 attributes. The complete list of queries used for person is presented in appendix B.

In figure 8.4 we present the distribution of the number of queries based on number of sources responding for each of them. As shown in the picture, the largest part of the queries had response from a small number of sources. This fact decreases the possibility of gathering possibly overlapping descriptions from multiple sources and may cause some bias in the learned rules. However, another explanation could also be related to the fact that three of the considered sources present a large number of descriptions about locations (Geonames, Okkam and Factual), whereas the other are more likely to refer to famous locations. In this case, the random selection of attributes proved to be less effective as preliminary blocking system due to the extremely large set of seed queries available. For this reason we decided to apply also a crawling-based approach looking at gathering possibly overlapping descriptions. The procedure described in section 8.2.1, basically consists in gathering descriptions from sources, using seed queries configured with the names of locations of descriptions gathered from a pivot resource. In figure 8.5 we present a view of the response size distribution for each of the queries. The largest amount of queries retrieved a relatively small amount of samples probably due to the high level of selectivity of the queries (e.g. Ban Tham). On the other side, a very small number of queries were so general to retrieve more than 50 description among all the sources. As shown in the graph, the large majority of queries retrieved allowed to retrieve less than 70 samples.

In table 8.2 we present a description of the data retrieved using this method. As it is possible to see the also location sources produced different results in terms of average number of attributes per description retrieved. Furthermore, some of the sources show how the average value is not reliably presenting a very high standard deviation. This aspect is quite representative of the structural heterogeneity and different of richness presented by different data sources. Also in this case we hope that this real life heterogeneity serves to learn matching rules suitable to be applied across these and many different other data sources about location.

Also the location dataset generated is very sparse with more than 3952 different attributes types, and only a relatively small set of attribute types shared among descriptions. This is mostly due to the large variety of sometimes very specific properties used in sources such as DBpedia and Freebase. We believe that this semantic heterogeneity is representative of the one affecting entity matching on the web, and thus it is suitable to support the learning of rules that should be applied in this context. This problem mostly challenges the semantic harmonization task, which becomes particularly cumbersome, as shown in section 7.2.2. The number of harmonized attributes

Source	#desc	#attr	avg. #attr	attr STD
dbpedia	809	50133	61.97	37.76
factual	7487	122277	16.33	7.59
freebase	1356	74629	55.03	64.49
okkam	275	2908	10.57	2.04
musicbrainz	453	115588	17.14	6.93
opencorporates	1534	22423	14.61	3.15
TOTAL	11914	277215	18.56	21.39

Table 8.3: Organization Dataset Collected Description

relying on defined mappings at the moment of writing is 459939 attributes, with 306169 attributes non harmonized. In average, every description presents 16.39 harmonized attributes, with a standard deviation of 13.06. The number of non harmonized attributes is in average 10.98, with a standard deviation of 37.29. The overall coverage ratio at the moment of writing is 0.60 for entity type location. This lower coverage ratio may also be related to the lower number of features related to the type location. Furthermore, locations descriptions are likely to contain a number of factual information not necessarily interesting for identification purposes such as rain in millimeter in the last year, or average hours of sun in month of may.

Organization Samples Collection

The queries to the data sources listed above are composed by randomly selecting words from a list of more than 22.000 names, obtained by processing the Person infobox dataset of DBPedia[13] looking for the names of organizations using the properties listed in table 8.4. Given a random name n, for each of the sources s considered, a search

[13] http://wiki.dbpedia.org/Downloads32#persondata

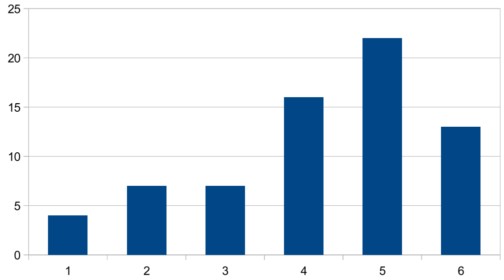


Figure 8.6: Number of queries per data sources returning samples

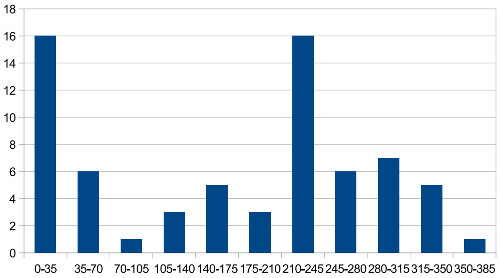


Figure 8.7: Response size distribution in blocks of samples retrieved

`http://dbpedia.org/property/formerAffliations`
`http://dbpedia.org/property/work`
`http://dbpedia.org/ontology/recordLabel`
`http://dbpedia.org/property/workplaces`
`http://dbpedia.org/property/affiliation`
`http://dbpedia.org/property/cteam`
`http://dbpedia.org/property/memberOf`
`http://dbpedia.org/ontology/team`
`http://dbpedia.org/property/group`
`http://dbpedia.org/property/currentTeam`
`http://dbpedia.org/property/debutteam`
`http://dbpedia.org/property/currentclub`
`http://dbpedia.org/property/politicalParty`
`http://dbpedia.org/ontology/party`
`http://dbpedia.org/property/currentteam`
`http://dbpedia.org/ontology/musicalBand`
`http://dbpedia.org/property/party`
`http://dbpedia.org/ontology/company`
`http://dbpedia.org/property/company`
`http://dbpedia.org/property/employer`
`http://dbpedia.org/property/workPlaces`
`http://dbpedia.org/ontology/workPlaces`
`http://dbpedia.org/property/workInstitution`
`http://dbpedia.org/ontology/employer`

Table 8.4: Properties used to gather organization names in persondata dataset

query $Qs(n)$ is submitted aimed at retrieving a list of identifiers. For each of these identifiers, a raw description is then retrieved from the considered source, and wrapped as an entity description $\mathcal{D}$, extended with contextual information, and stored together with the source URL and the name query used. We collected 11914 entities description with 69 seed queries (e.g. Champaign, Acadians, Schell, Atomic Kittens, Transylvanian, etc.). The description present a total number of 277215 attributes. The complete list of queries used for person is presented in appendix B.

In figure 8.6 we present the distribution of the number of queries based on number of sources responding for each of them. As shown in the picture, the largest part of the queries had response from at least 4 sources. This fact increases the possibility of gathering possibly overlapping descriptions from multiple sources. This fact is probably due to the low selectivity of some seed queries (e.g. university) that allowed to gather description from different sources. Differently from location, only two sources present a large number of descriptions about organization (OpenCorporates and Factual), whereas the other are more likely to refer to famous organization. Furthermore, both OpenCorporates and Factual focus on different types of organization. The former collects information about corporates, whereas the latter focuses more on restaurant, wine producers, and so on. In this case, the random selection of attributes can hardly be effective as preliminary blocking system due to the extremely large set of seed queries available. For this reason we decided to apply also a crawling-based approach looking at gathering possibly overlapping descriptions. The procedure described in section 8.2.1, basically consists in gathering descriptions from sources, using seed queries configured with the names of locations of descriptions gathered from a pivot resource. In figure 8.7 we present a view of the response size distribution for each of the queries. The number of selective queries that retrieved a small number of descriptions (up to 70) are balanced by a large number of queries that allowed to retrieve a relatively large amount of samples probably due to the low level of selectivity of the queries (e.g. Records, University, Brazilian, British, etc.).

In table 8.3 we present a description of the data retrieved using this method. As it is possible to see the also organization sources produced different results in terms of average number of attributes per description retrieved. Furthermore, some of the sources show how the average value is not reliably presenting a very high standard deviation. Also in this case, we interpret this heterogeneity as something quite representative of the structural heterogeneity and different of richness presented by different data sources.

Also the organization dataset generated is very sparse with more than 2218 different attributes types, and only a relatively small set of attribute types shared among descriptions. This is mostly due to the large variety of sometimes very specific properties used in sources such as Factual, DBPedia and Freebase. We believe that this semantic heterogeneity is representative of the one affecting entity matching on the web, and thus it is suitable to support the learning of rules that should be applied in this context. As for Person and Location, this problem mostly challenges the semantic harmonization task, which becomes particularly cumbersome, as shown in section 7.2.3. The number of harmonized attributes relying on defined mappings at the moment of writing is 221165 attributes, with 59591 attributes non harmonized. In average, every description presents 13.56 harmonized attributes, with a standard deviation of 12.83. The number of non harmonized attributes is in average 5.00, with a standard deviation of 10.54. The overall coverage ratio at the moment of writing is 0.73 for entity type organization. This coverage ratio may is very similar to the one defined for person. Probably, organization, as person, present in proportion a lower amount of attributes that are not relevant for information, as they present less statistical attributes that can be more interesting for locations.

8.2.2 Labeling Samples for Training Set

Once we collected data samples about specific entity types, we have the problem of labeling samples pairs to create a training set. This is a typical problem of supervised machine learning techniques [58, 67]. In fact, supervised machine learning methods are affected by matching rarity problem, which makes the problem of spotting positive matching samples particularly hard. This is particularly true for very large datasets. A common strategy applied to reduce the complexity of duplicate detection is to rely on a cheap similarity metric to cluster blocks of entities that are more likely to contain duplicates [79, 110, 53, 3]. Given the way the dataset is generated, we can rely on some metadata, such as provenance and keyword used to retrieve a description, to reduce the search space and ease the matching rarity problem. In doing this, we optimize the *Reduction Ratio* $RR = 1 - C/N$, where C is the number of candidate match, and N is the cardinality of the cross product of the datasets. Notice that there is a possibility of missing some matching sample considering the original datasets source because we rely on possibly imperfect lookup systems affecting $Pairwise\ CompletenessPC = S_m/N_m$, where S_m is the number of true matches among the candidates, and N_m is the number of true matches in the datasets [110]. Nevertheless, as long as the main goal is to select generic positive matching samples, rather than pairwise record linkage, this issue can be neglected for the moment. Michelson and Knoblock in [110] describe a very effective supervised blocking system. However, the supervised method relies on the existence of a training set. Thereby, the bootstrap of this method must rely on a distance metric, capable of ranking possible matches on base of a record similarity measure. In this work, we implemented a block extractor based on an Apache Lucene 3.4.0[14] inverted index and *tf-idf* similarity metric to define blocks of samples to be compared and labeled by a person. Given a set of descriptions D and a pivot description $d_i \in D$ about an entity, we use the complementary set of descriptions $D \setminus d_i$ to create an inverted index using the features of all the descriptions. Then, we select the *name* attributes of the pivot descriptions d_i to define a query to the inverted index. The query to the inverted index produces a ranked list of results, from which we select the top k results. We estimated $k = n + n * 0.3$, where $n = |S|$ is the number of sources considered plus an overhead considering possibly duplicates contained in the sources. Thus, assume that for person we relied on 7 different sources, then for each query we selected a top k list with $k = 7 + 0.3 * 7 = 9$. Furthermore, we decided to introduce some aleatory factors that should help in boosting ambiguous cases that are very important to create a training set suitable to learn effective entity matching rules. The aleatory factors

[14] http://lucene.apache.org/core/

foundation date: 2011-08-10 **organization type**: Private Limited Company **country**: United Kingdom **jurisdiction**: GB **has key people**: ALEXANDRA SIOBHAN LEEKS **name**: ANATHEMA LIMITED **foundation date**: 2011-08-10T20:06:13+01:00 **street address**: 30 BRICKENDON LANE, HERTFORD, HERTFORDSHIRE, ENGLAND, SG13 8HY

Table 8.5: Example of organization description extracted from OpenCorporates dataset

retrieved at: 2011-07-25T21:11:25+01:00 **organization type**: PRIVATE COMPANY LIMITED BY SHARES **jurisdiction**: GI **current status**: Struck Off By Request **company number**: 38853 **inactive**: false **updated at**: 2011-07-25T21:11:25+01:00 **name**: ANATHEMA LIMITED **foundation date**: 2011-03-22T21:07:51+00:00

Table 8.6: Example of organization description extracted from OpenCorporates dataset

considered are

- conjunctive boolean combination of the parts of names;
- randomized boost factor on random query tokens;
- string similarity metrics;

Thereby, given a description like the one presented in table 8.5, we produce a query to the index in the form of:

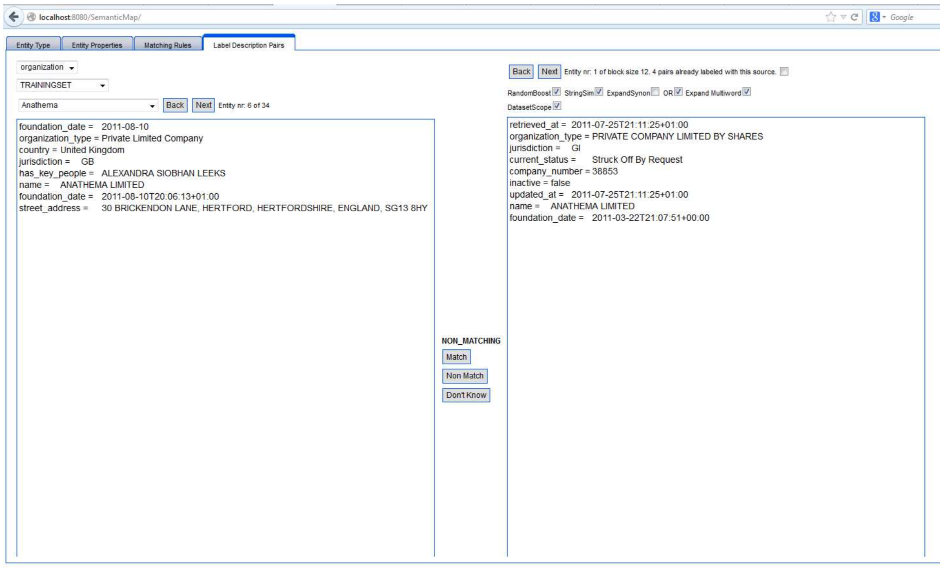
```
name:(
      ("ANATHEMA LIMITED") OR
      ("ANATHEMA"^0.5~0.9 AND "LIMITED"^5~0.9 ) OR
      ("ANATHEMA"~0.9 "LIMITED"~0.9 )
    )
```

Among other results, this query allowed to retrieve the sample of entity described in table 8.6. Relying on simple similarity metrics to block possibly similar entities is quite common in database record linkage [67, 127, 58]. However, among the existing methods, the inverted index was very convenient as allowed us to deal with the problem neglecting the structural heterogeneity affecting the descriptions we are dealing with. Furthermore, inverted index have also the advantage of granting sub-linear access time to the indexed data, allowing to create and query indexes dynamically.

We integrated the block extractor defined so far in a web application that would allow human user to take matching decision on the pairs of descriptions. The first version of the application, named SemanticMap, was developed relying on rich interface library Icesoft Icefaces 1.8.2[15] as J2EE Java Server Faces[16] 1.2 implementation. The samples labeled with matching decisions are then stored in a database and will then be used as a training set and evaluation set respectively. A screenshot of the application developed is depicted in figure 8.8. The next step is collect a large set of labeled samples aimed at capturing matching knowledge used by person in taking matching decision.

[15] http://www.icesoft.org/java/

[16] http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html



organization
TRAININGSET
Anathema
Entity nr: 6 of 34

foundation_date = 2011-08-10
organization_type = Private Limited Company
country = United Kingdom
jurisdiction = GB
has_key_people = ALEXANDRA SIOBHAN LEEKS
name = ANATHEMA LIMITED
foundation_date = 2011-08-10T20:06:13+01:00
street_address = 30 BRICKENDON LANE, HERTFORD, HERTFORDSHIRE, ENGLAND, SG13 8HY

Entity nr: 1 of block size 12. 4 pairs already labeled with this source. □

RandomBoost ☑ StringSim ☑ ExpandSynon □ OR ☑ Expand Multiword ☑
DatasetScope ☑

retrieved_at = 2011-07-25T21:11:25+01:00
organization_type = PRIVATE COMPANY LIMITED BY SHARES
jurisdiction = GI
current_status = Struck Off By Request
company_number = 38853
inactive = false
updated_at = 2011-07-25T21:11:25+01:00
name = ANATHEMA LIMITED
foundation_date = 2011-03-22T21:07:51+00:00

Figure 8.8: A screenshot of the Semantic Map application labeling pairs of samples.

The process proposed is the outlined in the figure 8.9. As previously mentioned, the rules learned must consider also the *unknown* cases as options for classification. Thereby, the user should be labeling samples considering three classes:

- Match: when the compared descriptions are recognized to be referring to the same real world entity without any doubt;
- Non Match: when the compared descriptions are recognized not to be referring to the same real world entity without any doubt;
- Dont Know: when the compared descriptions are too ambiguous to take any reliable matching decision;

We now formally define a training set sample σ as a triple $\sigma < \epsilon_1, \epsilon_2, \Delta >$ where, Δ is a matching decision, $\epsilon < i, s, q >$ is a tuple composed of i is a URI identifying an entity description $d \in D$ as presented above, $s \in S$ is the source of origin of the description, and q is the seed query used to retrieve such entity. Define a training set Γ as a set of labeled samples $\Gamma : \{\sigma_1, ..., \sigma_n\}$. Notice that the selection of a seed query as primary block source for building an index of retrieved descriptions as described in figure 8.9 does not necessarily hold if the samples collection process is the one based on iterative name-centered crawling. In fact, possibly matching descriptions may be indexed with different seed queries. This problem can be solved extending the scope of the indexed dataset beyond the seed query block. Namely, for each entity, we can index the whole dataset minus the pivot entity to be matched.

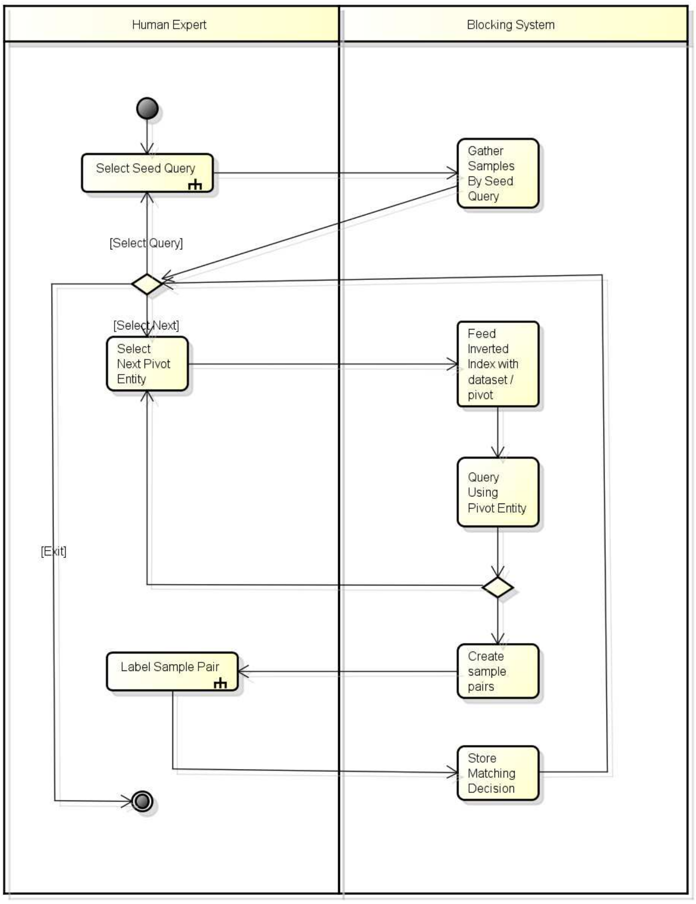


Figure 8.9: Activity Diagram of Training Set Labeling Process

As mentioned in the beginning of the chapter, we aim at capturing through machine learning techniques some sort of common knowledge applied by human users in matching entities in the context of the Web. For this reason, the samples must be labeled by several users that are not necessarily experts in the domain, but are capable of applying common sense knowledge related to what attributes have to match in two descriptions to take reliable matching decisions. In principle, this type of job is suitable to be crowdsourced [31]. There are approaches that aim at relying on a combination of crowdsourcing and automated methods, see for example [148]. However, reliable crowdsourcing is quite hard to obtain despite many problems can be easily presented in form of simple solvable tasks. In fact, despite platforms such as Amazon Mechanical Turk[17] provide access to a large number of tasks executors, a set of quality management issues persists [87], and often special purpose techniques have to be employed to discern usable results from garbage.

[17] https://www.mturk.com/mturk/welcome

We now have all the ingredients to create a training set suitable to extract matching rules suitable for an open environment:

- samples from different heterogeneous sources;
- an ontology presenting features about the entities of interest;
- an extensive list of mappings supporting the harmonization of the features towards the defined ontology;
- a method to block samples to be compared based on inverted index and *tf/idf*;
- a graphical interface to support labeling of pairs of descriptions;

The next step in line is to have several people labeling a relevant number of samples pair. Ideally, the training set should support the extraction of matching rules capturing the knowledge used by people while performing these type tasks. Initially, we planned to rely on an open platform such as the Amazon Mechanical Turk[18] as an open platform for crowd-sourcing this type of tasks. There exist solutions for relying on such platform to solve completely or partially the entity matching problem, e.g. [148]. However, there are several issues related with the adoption of this type of platform, which include, besides the costs, a serious commitment in the definition of cognitively sound experiments. This objective is quite far from our primary goal of proving the feasibility of the knowledge-based approach. Furthermore we rather would like to frame the solution around a community that would actually exploit the results of a knowledge-based solution for open entity matching and support its maintenance. The steps in this direction are going to be discussed further in the last part of this work. Thereby, we choose to collect samples in a controlled environment within in the Okkam Labs, in Trento. This choice has the two-fold advantage of granting sound labeling, and decreasing the complexity of the conduction of the labeling process. The main disadvantage of this approach, is that we can collect a relatively small set of samples from a limited number of person (around 10). We believe anyway that this factor is not relevant as what we seek for is an empirical evaluation that a knowledge-based solution which takes into consideration explicit semantic of attributes to take matching decision is suitable for an open, structurally and semantically heterogeneous environment.

The people that took part at the labeling process are:

- 3 women of different education background (cognitive science, political science and economy) and nationality (Italian and Brazilian);

[18] `https://www.mturk.com/mturk/welcome`

- 6 man of different education background (computer science, philosophy, human computer interaction) and nationality (Italian and Ukrainian);

the participants have an age in the range between 26 and 47 years, and all performed the labeling using their own computers at the times they wanted. The web application presented a option that would not propose twice the same sample pairs for labeling. However, this option could be arbitrarily switched off, and other users could cross check and change the evaluation of previously labeled samples. In order to have a sound, and as less as possible biased labeling process, we showed collectively few labeling samples to the participant and openly declared to them, that both matching and non-matching decision had to be sound and reliable in an open context. Thus, when a matching decision could not be taken reliably a don't know decision would have been appropriate. The execution of this type of task is prone to application of cognitive heuristics that would bias the results of the learning process. For this reason, we choose to display pairs of descriptions without respecting any particular order of attributes. On one side, this forces the person to scroll all the attributes looking for matching pairs and take reliable matching decisions. On the other side, this greatly increases the cognitive effort in taking matching decisions. A detailed description about the training set defined with the process described so far is presented in the following sections.

Training Set for Entity Type Person

Source	#desc	#attr	avg. #attr	attr STD
dbpedia	500	40343	80.68	35.63
factual	276	5959	21.59	4.34
freebase	417	28059	67.28	59.31
lastfm	207	6615	31.95	15.52
musicbrainz	321	3005	9.36	6.05
okkam	273	3252	11.91	1.11
opencongress	100	1521	15.21	4.76
TOTAL	2094	88754	33.03	35.93

Table 8.7: Person Training Set Description

For the entity type Person 7405 sample pairs were labeled, involving 2094 different descriptions and producing 549 positive matching labeled pairs, divided in 337 clusters distributed as depicted in figure 8.11. A further detail about the distribution of the cluster of positively labeled samples in terms of the data sources where the data samples were collected is presented in the graph 8.10. The training set contains also 6024 negatively labeled sample pairs, and 832 samples pairs labeled as don't know. The

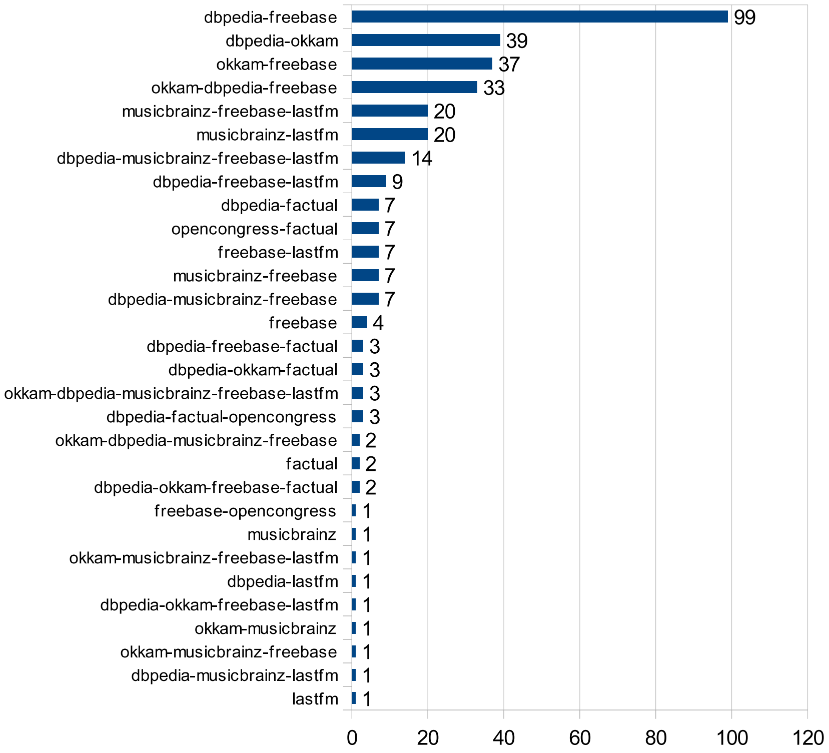


Figure 8.10: Distribution of clusters in terms of their sources

don't know samples pairs are very important, as they represent those descriptions that contain particularly ambiguous descriptions. Some statistics about the quality of the descriptions composing the training set are presented in table 8.7. The mappings for semantic harmonization covered the 73% of the overall attributes, with an average of 24 attributes harmonized per description, and a standard deviation of 27.

In figure 8.12 we present a graph outlining the distribution of the training set samples with respect to the pairs of data sources considered. As is possible to see there is a large number of non-matching sample pairs involving musicbrainz, freebase, dbpedia and okkam. The large amount of negative samples involving musicbrainz can be easily explained by the fact that this data source is quite extensive but presents a very vertical type of entity (i.e. musical artist). Thus, the blocking system defined proposed for labeling a large amount of sample pairs presenting names of artists with small variations in the name. The collection of such samples is probably also the result of the usage of not selective queries that allowed to gather of large amount of musicbrainz samples. The positive matching samples pairs are smaller in number with respect to negative

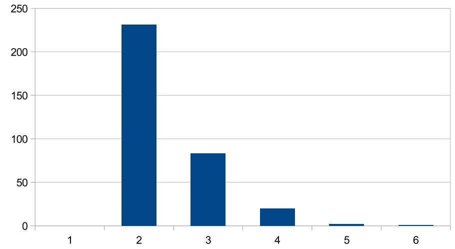


Figure 8.11: Cluster size distribution in training set for Person

sample pairs quite well distributed across the source pairs. The largest amount of positive matching samples are between freebase and dbpedia, followed by okkam and freebase and okkam and dbpedia. This is probably due to the fact that all the three data sources present information about mostly famous people. Another consistent set of positive matching sample pairs is related to the music artists. In fact there is a good number of positive matching samples between freebase, lastfm, musicbrainz and okkam. Very little contribution in terms of positive matching sample pairs is coming from factual and opencongress. These data sources presented very vertical dataset about health care practitioners, athletes and us politicians. Some of the descriptions gathered overlapped with the most famous entities contained in the other sources, but in general the amount of positive matching samples is little. We will analyze through experiments what is the impact of such heterogeneous distribution of data samples, applying filters and partitioning the dataset as described in section 8.2.3 and 8.2.6.

Training Set for Entity Type Location

Source	#desc	#attr	avg. #attr	attr STD
dbpedia	91	6672	73.31	39.27
factual	238	5754	24.17	69.43
freebase	289	15326	53.03	70.42
geonames	120	2006	16.71	4.22
okkam	275	3607	13.11	5.39
TOTAL	1013	33365	25.95	50.34

Table 8.8: Location Training Set Description

For the entity type Location 2310 sample pairs were labeled, involving 1013 different descriptions and producing 128 positive matching labeled pairs, divided in 90 clusters distributed as depicted in figure 8.14. A further detail about the distribution of the cluster of positively labeled samples in terms of the data sources where the data samples

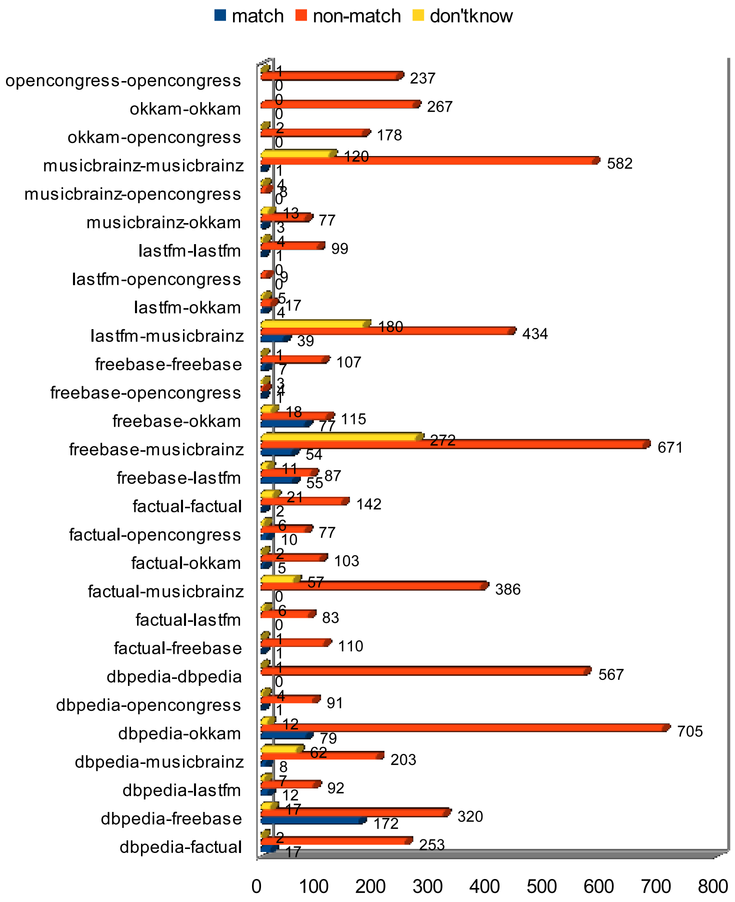


Figure 8.12: Distribution of samples per pairs of resources for Person

were collected is presented in the graph 8.13. The training set contains also 2119 negatively labeled sample pairs, and 63 samples pairs labeled as don't know. Also in this case, the don't know samples pairs are very important, as they represent those descriptions that contain particularly ambiguous descriptions. Some statistics about the quality of the descriptions composing the training set are presented in table 8.8. The mappings for semantic harmonization covered the 58.10% of the overall attributes, with an average of 15 attributes harmonized per description, and a standard deviation of 12.

In figure 8.15 we present a graph outlining the distribution of the training set samples with respect to the pairs of data sources considered. As is possible to see there is a large number of non-matching sample pairs involving musicbrainz, freebase, dbpedia

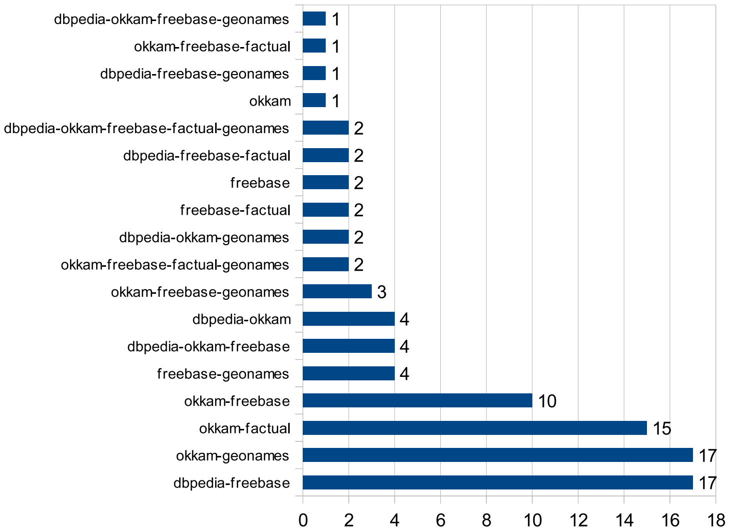


Figure 8.13: Distribution of clusters in terms of their sources for Location

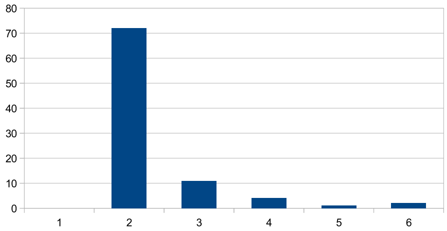


Figure 8.14: Cluster size distribution in training set for Location

and okkam. Differently from the case for person, a large amount of negative matching labeled sample pairs is individualized within the same sources. In fact, both okkam, factual and freebase present a large number of negatively labeled samples. This could be the effect of the fact that many queries gathered a small amount of descriptions from a small number of sources as presented in figures 8.4 and 8.5 at page 153. Thus, the system defined to gather descriptions combined with a smaller number of labeled samples produced a large number of negative samples collected from the same data source. Furthermore, a large amount of queries were too selective, gathering up to 23 descriptions in all. However, the absolute largest number of negative matching sample pairs is between okkam and factual. The positive matching samples pairs is relatively small, but well distributed across the source pairs. The largest amount of positive matching samples are between freebase and okkam, followed by dbpedia and freebase

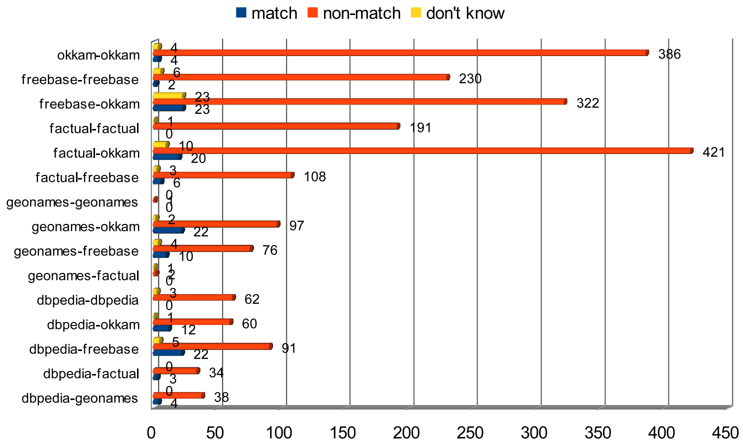


Figure 8.15: Distribution of samples per pairs of resources

and geonames and okkam. This is probably due to the fact that all the three data sources present overlapping set of descriptions about famous location. We will analyze through experiments what is the impact of such heterogeneous distribution of data samples, applying filters and partitioning the dataset as described in section 8.2.3 and 8.2.6.

Training Set for Entity Type Organization

Source	#desc	#attr	avg. #attr	attr STD
dbpedia	345	26609	77.12	38.32
factual	1501	22862	15.23	5.6
freebase	509	27701	54.42	70.55
musicbrainz	160	1856	11.6	8.03
okkam	85	990	11.64	2.14
opencorporates	451	6563	14.55	3.18
TOTAL	3051	86581	22.11	29.18

Table 8.9: Organization Training Set Description

For the entity type Location 5064 sample pairs were labeled, involving 3051 different descriptions and producing 177 positive matching labeled pairs, divided in 127 clusters distributed as depicted in figure 8.17. A further detail about the distribution of the cluster of positively labeled samples in terms of the data sources where the data samples were collected is presented in the graph 8.16. The training set contains also 2119 negatively labeled sample pairs, and 63 samples pairs labeled as don't know. Also in

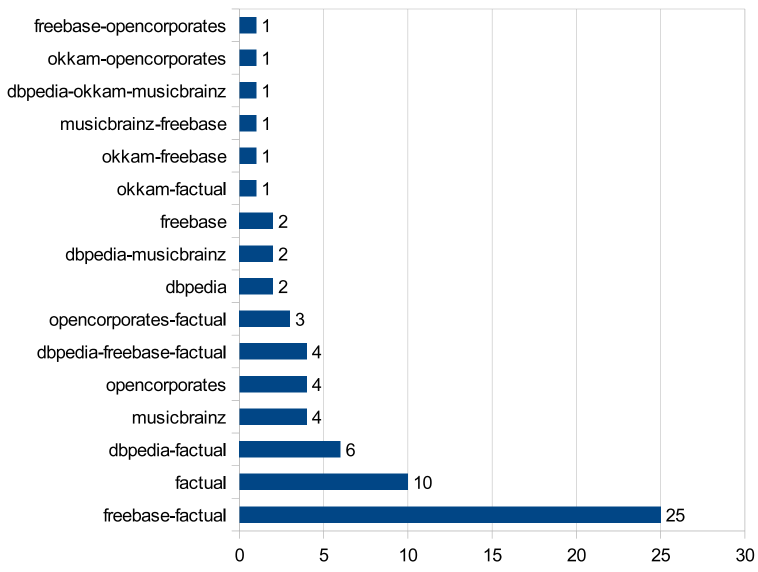


Figure 8.16: Distribution of clusters in terms of their sources for Organization

this case, the don't know samples pairs are very important, as they represent those descriptions that contain particularly ambiguous descriptions. Some statistics about the quality of the descriptions composing the training set are presented in table 8.8. The mappings for semantic harmonization covered the 73.62% of the overall attributes, with an average of 16.2 attributes harmonized per description, and a standard deviation of 17.76.

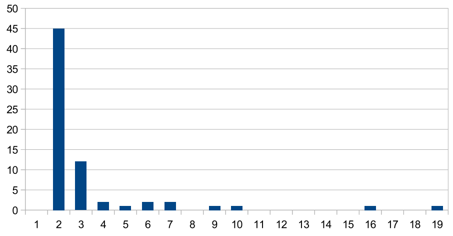


Figure 8.17: Cluster size distribution in training set for Organization

In figure 8.17 we present a graph outlining the distribution of the training set samples with respect to the pairs of data sources considered. As is possible to see there is a large number of non-matching sample pairs involving musicbrainz, freebase, dbpedia and okkam. Similarly for what happened with location, a large amount of negative matching labeled sample pairs is individualized within the same sources. In fact, both

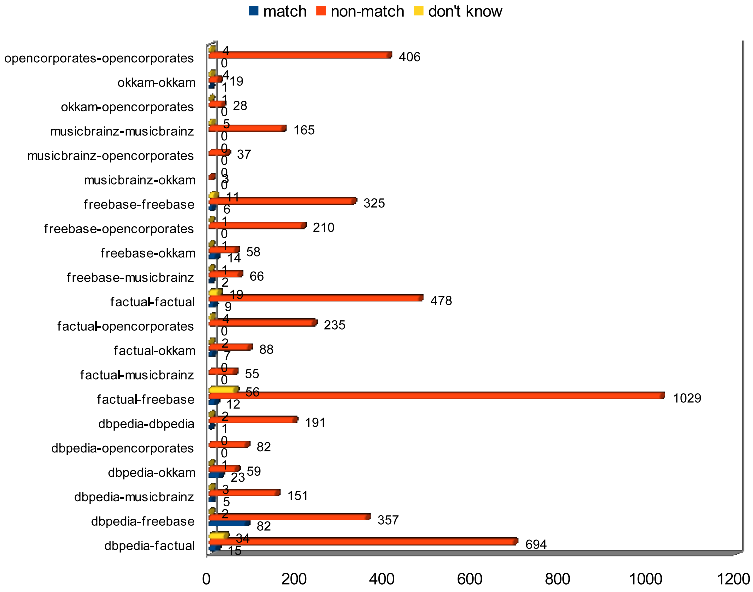


Figure 8.18: Distribution of samples per pairs of resources

factual, opencongress and freebase present a large number of negatively labeled samples of the same source. However, this time the effect should not be due to the query to the fact that many queries gathered a small amount of descriptions from a small number of sources. In fact, in average queries gathered a large number of samples in average from 4 to 6 sources in figures 8.6 and 8.7 at page 155. Our interpretation of this fact is that factual, freebase and opencorporate returned a much higher number of samples. Furthermore, opencorporates and factual contain a large number of non-famous entities such as wine producers, restaurants for factual, and in general corporates for opencorporates. These specialization allowed to gather a large number of samples that are not shared with other sources. This created some sort of spiral for the person labeling the samples when defining blocks of entities to be compared. Opencorporates, in particular, contained information about many companies opened with a name and a sequential number. Thus, this time the fact that many negative matching samples pairs are collected from the same sources is due to a combination of the quality of data collected and the the labeling process covered just a part of the data collected. The positive matching samples pairs is relatively small, but well distributed across the source pairs. The largest amount of positive matching samples are between freebase and dbpedia, followed by dbpedia and okkam and dbpedia and factual. This is probably due to the fact that all the three data sources present overlapping set of descriptions about famous organizations such as universities and large corporation. We will analyze through experiments what is the impact of such heterogeneous distribution of data samples, applying filters and partitioning the dataset as described in section 8.2.3 and 8.2.6.

8.2.3 Training Set Filter

Supervised machine learning techniques are strongly affected by the training set used to train them [67]. Ultimately, training processes are affected by the number of samples presenting some characteristics, relying on different approaches to estimate and treat outliers. Furthermore, many techniques rely on sets of methods to reduce over-fitting problems `supervised-machine-learning:a-review-2007`. Namely, they adopt techniques to avoid learning classification features too specific of the training set that would poor prediction in a more general context. Hence, the quality of the training set directly affects the quality of the classification results. For this reason we define a set of filter functions that allow us to remove matching samples that are not considered useful for classification purposes, or that affect negatively the learned classifier. We are aware of the fact that we are introducing a bias in the learning process, and for this reason, we plan to experimentally evaluate the impact of the adoption of such filters.

Let's define a filter function $\phi : \Gamma \rightarrow (\Psi \rightarrow B) \rightarrow \Gamma$ as a recursive function filtering labeled samples from a training set Γ to based on boolean function $\psi : \Sigma \rightarrow B$ where $B = \{True|False\}$:

$$\phi(\Gamma, \psi) : \begin{cases} \{\}, & \text{if } \Gamma \text{ is empty} \\ \phi(\Gamma^{i-1} \cup \sigma^i, \psi), & \text{if } \psi(\sigma) = False \\ \phi(\Gamma^{i-1}, \psi), & \text{otherwise} \end{cases} \tag{8.1}$$

The filter function defined above can be seen as some sort of second order function that applies the filter business logic defined by the boolean function ψ. ψ functions can implement any type of filtering business logic, including both syntactical filters on a specific type of attribute, cardinality filter related to the number of attributes composing descriptions, etc. In the following sections, we will briefly describe few of them we believe could impact the rule learning process.

Minimum Attribute Number Filter

An example of filter that could applied to a training set used to learn entity matching rules is a minimum attribute number filter. Namely, we would want to filter from the training set the samples containing descriptions with a number of attributes below a

1	**name**: Ferrero, Martin; **name**: Martin Ferrero; **subject**: Category:Miami Vice; **birthPlace**: Brockport, New York; **placeOfBirth**: United States; **subject:Category**: American film actors; **placeOfBirth**: Brockport, New York; **birthYear**: 1947; **birthPlace**: United States; **birthPlace**: U.S.; **label** : Martin Ferrero; **dateOfBirth** : 1947-07-13; **givenName** : Martin; **surname** : Ferrero;
2	**short_description**: Martin Ferrero (born July 13 1947 in Brockport, New York)is an American stage and film actor known for acting in the movie Miami Vice.; **first_name:** Martin;

Table 8.10: Examples samples filtered using minimum number of attributes in description filter function, with k=3

certain threshold. Lets define $\phi_{MC} = \Gamma \times N \rightarrow \Gamma$ as the function that applies on every sample pair $\sigma \in \Gamma$ the boolean function $\psi_M C : \Sigma \times N \rightarrow B$ defined as follows:

$$\psi_{MC}(\sigma, k) : \begin{cases} true, & \text{if } \exists e \in \sigma \wedge d \in e \wedge |d| < k \\ false & \text{otherwise} \end{cases} \tag{8.2}$$

Now we can define

$$\phi_{MC}(\Gamma, k) : \Gamma \setminus \{\sigma \in \Gamma | \psi_{MC}(\sigma, k)\} \tag{8.3}$$

and k is the minimum number of attributes required in a description not to be filtered. The training set then becomes $\Gamma_{MC} = \phi_{MC}(\Gamma)$. This type of filter could help in reducing the effect of structural heterogeneity in learning the rules for entity matching. Namely, assume that there are two descriptions, one presenting a whole list of attributes, and the other presenting the same attributes in the one unique textual paragraph. For a human user it is easy to take matching decisions as a person can easily interpret both structured and unstructured information to take matching decision. However, these type of samples can negatively affect the capability of learning useful rules as the classifier would not be able to unfold the information presented in natural language and thus the sample would bias oddly the classification process both for positive and negative matching samples. An example of descriptions that should be filtered is presented in table 8.10.

Minimum Shared Attributes Type Filter

Following an intuition similar to the one presented in section 8.2.3, we may further want to further filter labeled samples with descriptions that do not present a sufficient number of attributes types overlapping. This type of filters aims at removing samples presenting attributes that cannot be compared due to problems related to structural heterogeneity of the descriptions. This filter is subsumed by the Minimum Attribute Number Filter described in section 8.2.3. In fact, this filters removes samples presenting a large set of different attributes, but possibly contains information leading to matching decision into textual paragraphs. Again, a person is capable of interpreting both types of information, but this interpretation could not be captured by the classifier.

Define $n : D \rightarrow \{A\}$ is a function that given a description $d \in D$ returns the names $\alpha \in A$ of all features present in the description. Now we have to defined a boolean function $\psi_{MSC} : \Sigma \times N \rightarrow B$ that taken a sample pair $\sigma \in \Gamma$ and an integer k, returns true if the number of attributes shared between the descriptions mentioned in σ is below a threshold k. More formally:

$$\psi_{MSC}(\sigma, k) : \begin{cases} true, & \text{if } e_1, e_2 \in \sigma d_1 \in e_1 \wedge d_2 \in e_2 \wedge |n(d_1) \cap n(d_2)| < k \\ false & \text{otherwise} \end{cases} \tag{8.4}$$

Define the function $\phi_{MSC} = \Gamma \rightarrow \Gamma$ as the function that filters from

$$\phi_{MSC}(\Gamma, k) : \Gamma \setminus \{\sigma \in \Gamma | \psi_{MSC}(\sigma, k)\} \tag{8.5}$$

and k is the minimum number of shared attributes between the compared descriptions. An example of samples that should be filtered using the minimum shared attribute number filter is presented in table 8.11. This is done also following the intuition that if samples share a higher number of attributes, the training set may need to be split on the more attributes to classify correctly the samples, increasing the hight of the decision tree learned, and thus the length of learned rules. This principle is mentioned also in [161], where cascade combinations of decision trees is optimized by learning higher decision tree classifiers (i.e. tree with average longer path root-to-leaves). On the other side, considering shorter rules would ensure a higher applicability of the rules, but we are likely to pay a cost in terms in precision of the matching decision. However, also this aspect should be experimentally evaluated.

1	**name**: Ferrero, Martin; **name**: Martin Ferrero; **subject**: Category:Miami Vice; **birthPlace**: Brockport, New York; **placeOfBirth**: United States; **subject:Category**: American film actors; **placeOfBirth**: Brockport, New York; **birthYear**: 1947; **birthPlace**: United States; **birthPlace**: U.S.; **label** : Martin Ferrero; **dateOfBirth** : 1947-07-13; **givenName** : Martin; **surname** : Ferrero;
2	**short_description**: Martin Ferrero (born July 13 1947 in Brockport, New York)is an American stage and film actor known for acting in the movie Miami Vice.; **first_name:** Martin; **family_name:** Ferrero; **lives_in:** Los Angeles; **favoriteFood:**Pizza; **marriedTo:** Nice Lady; **favoriteMovie:** Indiana Jones 2; **studiedAt:** Cool Actors Acting School;

Table 8.11: Examples samples filtered using minimum number of shared attributes in sample filter function, with k=3

8.2.4 Learning a Decision Tree

Decision tree is a simple and quite effective classifier, or predictive model. A decision tree presents as nodes the attributes used in the training samples, and as leaves the target classification classes (e.g. Match, Non Match). Decision Tree can be induced from a training set using C4.5 algorithm [124]. C4.5 is a recursive algorithm based on the estimation of the Information Gain for each of the attributes[113]. Information gain relies on the estimation of the Information Entropy, a concept coming from Information Theory. Information Entropy essentially measures the uncertainty related to a random variable. For example, if we toss a fair coin, the entropy of each toss is at its maximum, i.e. 1. On the other side, if the coin is not fair, and the probability of getting each of the faces is not exactly 50%, then each toss will have a more easily predictable outcome, and thus an entropy value below 1. At its extreme, if the coin presents the same symbol on both faces, then the entropy of each toss will be zero, as we will precisely predict the outcome of the toss with 100% precision. So, in a sense, entropy is somehow a measure of impredicability. Information Gain (IG) is intuitively the expected difference in information entropy for each decision represented in the decision tree, i.e. $IG = H(T) - H(T|a)$. The attribute a with the highest information gain (i.e. the lower overall information entropy) is chosen recursively to split the training set and thereby to build the decision tree. The C4.5 algorithm is described below. For each of the base-cases:

- All the samples of the list belong to a class. Creates a leaf node for that class.
- No information gain, creates a node on the upper level.
- Instance of previously unseen class, creates a node on the upper level.

```
buildTree(Tree t, samples){
  check base cases
  for each attribute a{
      listIG.add(IG(a));
  }
  best_a = max(listIG);
  T1 = addnode(best_a,T);
  buildTree(T1,samples);
}
```

For a complete description of the algorithm, please refer to original work of Quinlan [124].

Learning of decision tree based on the training set as defined in section 8.2.2 was executed relying on Weka 3.6.5 [106]. Weka data mining software[19] is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied

[19] http://www.cs.waikato.ac.nz/ml/weka/

directly to a dataset or included in Java code as Apache Maven[20] components. In particular, the learning of decision tree was executed relying on the *J.48* implementation of the C.45 algorithm [124] described above.

The first necessary step to learn a decision tree is to represent the pairs of labeled descriptions in a way that can be understood and consumed by the learning algorithms. In particular, what is necessary is to produce a comparison vector $v : [w_1, ..., w_n]$ where w_i represent the similarity score of the strings representing the value of the attributes α_i contained in both the descriptions. Practically, the attributes of the same type in the descriptions are compared, according to some comparison operator κ, and score result of the comparison is used to witness how similar are the values of in the descriptions for that specific attribute type. Recalling the definition of (6.5), κ is as a comparison operator that taken two strings $s_1, s_2 \in \mathcal{S}$ returns a similarity measure between 0 and 1 according to a similarity metric $\omega \in \Omega$. Thereby, in order to produce a similarity vector v, we sort all the features defined for the compared type, and we produce a series of similarity scores obtained by applying a function κ that given the different instances of the same attribute type produces a similarity score. The combination of the scores of all the matching attributes, combined with the matching decision associated with descriptions, produces an entry for the learner. When an attribute appears in only one of the two descriptions, or does not appear in any of them, it is clearly not possible to produce any similarity score, and thus the value in the similarity vector is set with a special character to highlight the 'unknown' value for that specific attribute.

It is clear that the definition of the similarity vector is strongly affected by two factors:

- the similarity metric adopted;
- the comparison method adopted;

The similarity metric estimates the similarity of attributes when these are represented with syntactic variations across different datasets. This problem was deeply studied in the past, and one single solution satisfying all of the cases was not found. It seems intuitive to assume that some similarity metrics work better than others according also to the type of attribute. These and other considerations about string similarity metrics are analyzed more in depth in section 9.1.

The method of comparison adopted also strongly affects both learning and matching process. In fact, when a description contains more than one attribute of the same type, it becomes problematic to choose which one should be representative for the comparison of that pair of descriptions. In fact there exists attributes that are meant

[20] `http://maven.apache.org/`

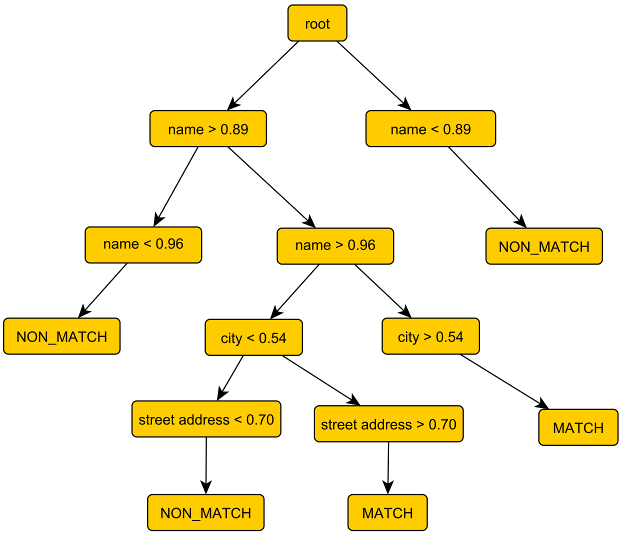


Figure 8.19: Example of a decision tree extracted from Organization training set

to be instantiated several time with different values. Consider for example the property *contained by* which enumerates the transitive hierarchy of locations containing another location. Another example could be the property *domain tag*, that enumerates in different attribute instances a list of keywords about the person, organization or location. In other cases, simply there exists legitimate variations of the same value of an attribute (e.g. U.S. and United States). Several consideration about how to deal with this type of issue is presented also in section 9.2.

8.2.5 Extracting Rules From Decision Trees

In the following we briefly describe a method for extracting rules for open entity matching given a decision tree. For the moment, we consider a single training set Γ from which it is possible to extract a single decision tree classifier. A decision tree can be represented as finite tree hierarchy, namely a collection of nodes related by parent/child relation. All nodes have a unique common ancestor named root. Every node, starting from the root, can have an arbitrary number of child nodes. The height of a tree (or depth) is the number of nodes that have to be traversed to reach a leaf starting from the root. Child nodes with a common parent are sibling nodes. The leaves of a tree are nodes with no child. Every path from the root of the tree, to each of the leaves corresponds to a matching rule. The leaf of a decision tree learned using a classifier corresponds to the class to which a sample satisfying all the element of the rule belongs. The process of extracting production rules from decision three is described in details in [123]. A classifier $c \in C$ produces a decision tree, from which is possible to extract a set of rules $\mathscr{P}$ as defined in section 6(eq. 6.1).

So far, we defined matching rules, a set of tools to verify their satisfaction given pairs of descriptions, and a standard process for extracting rules from decision trees. In [123] was shown how relying on production rules extracted from decision trees it is possible to achieve improvements in performances, maintaining the advantages of relying of explicit and easy to understand set of rules. However, relying on rules extracted from a single decision tree may be limiting. In fact, decision tree learning may not be optimal in very large and heterogeneous contexts. Learning a decision tree has a high computational complexity as it belongs to the NP-Complete class of computational complexity [85]. Hence, most algorithms including C.45 described in section 8.2.4, rely on greedy heuristics to take optimal local decisions, but do not guarantee optimal global solution. Furthermore, as many machine learning techniques, decision tree learner suffer of possible problems of over-fitting compensated by pruning heuristics [115]. However, pruning may end up marginalizing not only spurious samples, but also relevant samples only marginally represented in the training set. These facts suggest that pursuing rules extractions from a unique and large training set can be sub-optimal, and thus alternative strategies should be explored. Among others, in this thesis we explore the possibility of obtaining entity matching rules extracted from decision trees learned from partitions of the training set obtained applying several combinatorial heuristics. These partitions of the training set are obtained by applying filter functions to samples composing the training sets. Filtering functions were presented more in detail in the section 8.2.3. In section 8.2.6 we present describe some method of training set partitioning based on filter function, and propose some combinatorial heuristics. The combination of rules learned from different decision trees creates the problem of merging rules. This aspect is treated more in detail in section 6.4.

8.2.6 Training Set Partitions and Combinations

In this section we propose few heuristic principles to partition the training set to support the learning of effective and reliable entity matching rules. First of all, we have to define the dimensions we consider relevant for this goal. In this context, we propose to partition the training set according to:

- *source of the description.* The training set should be partitioned taking into consideration the data source from which the description was retrieved. This dimension is useful to create partitions that are the result of combinations of different sources, without having to consider the whole dataset at once. The intuition behind this partition choice is that some data sources may contain specific type entities (e.g. about musicians), but if the number of samples matching these entities is too small, then specific features for matching musicians may get lost. Thereby, considering pairs, or combinations of data sources seems a lead worthy to be experimentally evaluated.

- *class of classification.* The training set should be partitioned taking into consideration the class of classification of the used samples. As mentioned in section 8.2.2, we ask people to label pairs of description considering three classes: Match, Non Match and Don't Know. Multi-class decision tree classifiers may be affected by higher error rate in training with respect to simple binary classification [1]. However, it is not convenient to give up the information related to uncertainty in the classification contained in the Don't Know labeled cases. Hence, partitioning the training set combining samples of different classes seems a lead worthy to be experimentally evaluated.

Assuming that we more than two data sources, we propose to define training set combining the partitions of labeled pairs of descriptions collected from combination of data sources. In particular we consider:

- *data source binary combination.* We produce partitions of the training set considering all possible combination of data sources of cardinality 2. For example, considering data sources A, B and C, we produce training set partitions containing labeled samples collected from (A,B), (B,C) and (A,C).

- *data source power set.* We produce partitions of the training containing considering the power set combination of the data sources available. Given a set of data sources S, we compute 2^S, excluding empty set and all the subsets with cardinality lower than 2. For example, considering data sources A, B and C, we produce training set partitions containing labeled samples collected from (A,B), (B,C) and (A,C) and (A,B,C).

Considering that the number of classification classes is stable at 3, we consider three different types of partitioning:

- *simple binary.* We produce a training set for classification considering only samples labeled as Match and Non Match, ignoring the Don't Know cases.

- *binary combination.* We produce 3 training set for classification considering pairs of classes ($Match$,$NonMatch$), ($Match$,$DontKnow$), ($NonMatch$,$DontKnow$). The intuition underlying this training set configurations is to explore at its best the capability of the Don't Know samples of capturing uncertain classification, optimizing the precision of learning Match and Non Match classification

- *powerset combination.* We produce 4 training set for classification considering combination of the powerset of the classes ($Match$,$NonMatch$), ($Match$,$DontKnow$), ($NonMatch$,$DontKnow$) and ($Match$,$NonMatch$,$DontKnow$), excluding empty set and all the subsets with cardinality lower than 2. Given the small number of classes, this type of combination will be experimentally tested for the sake of completeness.

At this point, we outlined 6 possible variants of learning process that can in principle produce diverse and overlapping sets of rules. In fact, partitioning the training sets to train several classifiers are going to produce several sets of rules $\{\{\rho_{11}...\rho_{1n}\}, ..., \{\rho_{m1}...\rho_{mk}\}\}$, each of them containing rules. The main objective of this process is to obtain a set of rules more extensive and more precise than the one obtained relying on a single global training set. However, we are likely to find a large set of rules that need to be merged. In the following section we formally define the normalization and merging rule processes.

8.3 Combining Top-down and Bottom Up Rules

In this section we aim at describing possible strategies for the integration of bottom-up learned rules as described in section 8.2 with the rules result of an ontological analysis of the properties as presented in section 8.1. The optimal integration of these rules must provide improvements with respect to the adoption of each of the set of rules considered separately. Intuitively, combining rules result of different and complementary processes should not be complicated. However, if we put together both rules as they are, we are likely to find some inconsistencies. These would mostly concern the similarity thresholds used to evaluate the satisfiability of rules containing atoms with features appearing both in the bottom up and top-down rules. In fact, top-down rules are defined without considering neither any specific similarity metric, nor a similarity threshold coping with possible misspelling or syntactic variants. Therefore, in the reminder of the section we propose several strategies, each of them with pros and cons, that must be experimentally evaluated.

8.3.1 Plain Rules Combination

A first, simple baseline approach is to extend the set of rules result of the learning process, and then merge them when possible. This way, the top-down rules have the same relevance than the learned rules, and they are treated exactly as if they were the result of some learning process. Thus, let $\mathcal{P}_b$ be the set of rules learned in a bottom-up fashion, and $\mathcal{P}_t$ be the rules result of a top-down process. Plain rules combination process consists in defining a new set of rules $\mathcal{P}_i = \mathcal{P}_b \cup \mathcal{P}_t$ without performing any further merging or normalization on the similarity thresholds.

In order to evaluate the impact of the integration of matching rules obtained relying on meta-properties of attributes defined in the identification ontology, we need to perform experiments combining rules learned from data with positive only and negative only top down matching rules.

8.3.2 Plain Top Down Threshold Normalization

A second, simple approach we propose is to perform a plain integration of the rules as described in section 8.3.1, performing a threshold normalization as described in section 6.4.3. The idea is that top-down rules and bottom-up rules should be preserved without performing any merging, but when possible we should consider modifying top-down rules atom to embed similarity thresholds result of the learning process. Top-down rules are the result of ontological analysis that does not take into consideration syntactic representation of data, and thus combining these two aspects should provide a set of rules embedding matching exploiting the points of strength of both approaches.

Thus, let $\mathcal{P}_b$ be the set of rules learned in a bottom-up fashion, and $\mathcal{P}_t$ be the rules result of a top-down process. Plain rules combination process with top down threshold normalization consists in defining a new set of rules $\mathcal{P}_i = \mathcal{P}_b \cup \mathcal{P}_t$ performing for example a *Relaxed Match and Conservative Non Match* (i.e. RC defined in section 6.4.3) cross rules threshold normalization that would affect all the atoms of top down rules contained in both top-down and bottom-up rules. This operation would consist in the normalization of top-down defined positive matching rules to rely on a more relaxed threshold for positive matching decisions, and the normalization of defined negative matching rules to rely on a more conservative threshold for negative matching decisions.

The expected effects of this normalization step is to smooth the application of top-down rules to accommodate syntactic heterogeneity without loosing the constraining power of top-down rules. Furthermore, we intend to evaluate the impact of the integration of normalized matching rules obtained relying on meta-properties of attributes defined in the identification ontology by performing experiments combining rules learned from data with positive only and negative only top down matching rules.

8.3.3 Top-Down Priority Rules Combination

Another approach could be to consider top-down defined rules with a higher relevance with respect to the bottom-up learned rules. This relevance could be interpreted defining a formal hierarchy of the provenance of rules implemented through the subsumption mechanism, and defining a new merging function that allows to merge rules obtaining more general rules. Namely, we assume that ontological knowledge expressed through meta-properties of attributes is generally valid and thus we choose to endorse this type of rules on place of the one learned in a bottom-up fashion. From a practical perspective, this corresponds to the choice of merging any rule subsumed by a top-down extracted rule towards the latter. Let $\mathscr{P}_b$ be the set of rules learned in a bottom-up fashion, and $\mathscr{P}_t$ be the rules result of a top-down process. Formally, we define $\mu\rho TD : \mathscr{P} \times \mathscr{P} \to \mathscr{P}$, as the merging function::

$$\mu_{\rho TD}(\rho_1, \rho_2) = \begin{cases} \rho_1, & \text{if } (\rho_2 \sqsubseteq_\rho \rho_1) \wedge \rho_1 \in \mathscr{P}_t \\ \rho_2, & \text{if } (\rho_1 \sqsubseteq_\rho \rho_2) \wedge \rho_2 \in \mathscr{P}_t \end{cases} \tag{8.6}$$

Assuming that the subsumption-based merging process described in section 6.4.4 was already executed on bottom-up extracted rules.

8.3.4 Positive and Negative Only Top Down

Another approach could be to consider only positive or negative matching top-down defined rules as an integration of bottom-up learned rules. In fact, one could rely on knowing that the training set supporting bottom-up rules is largely formed by negative matching samples. This could imply that negative matching rules are well represented, and no further integration is necessary. Therefore, we have to evaluate the impact of the integration of top-down positive matching rule only, considering plain integration and integration with threshold normalization.

Chapter 9

Fingerprint Match Solution

Given the premises outlined in the second part of the thesis, and the vision presented in chapter 4, it is clear that the main challenges for the realization of a reliable and effective Knowledge-Based Entity Matching algorithm are:

1. the definition of adequate identification ontology, encompassing the attribute types that are relevant for taking open world matching decision;
    (a) selection of the entity types considered;
    (b) selection for each of the entity type of a set of properties suitable to be adopted as identity criteria along matching process;
    (c) annotation of each of the properties with meta-properties supporting the elicitation of matching rules as result of ontological analysis;
2. the definition of an effective semantic harmonization process mapping equivalent concepts and attributes to the one defined in the identification ontology;
    (a) production of contextual mappings for entity types towards known ontologies and schemas;
    (b) production of contextual mappings for features towards known ontologies and schemas;
3. the definition of some heuristic easing the problems related to structural heterogeneity exploiting knowledge about the semantic of the considered features;
4. define a set of identification rules that can guarantee precise matching decision in an open world context;
5. define a matching process suitable to combine all the matching knowledge to take reliable matching decisions when possible;

Point 1 was treated in detail in chapter 5, together with a formalization of the point 2 in section 5.6. In chapter 7 we presented some solutions to the problems related to semantic and structural heterogeneity issues. Precisely, in section 7.1 we briefly described the process producing mappings of known entity types towards the types defined in the identification ontology (point 2.1). In section 7.2 we present in detail the mappings defined for the features of each of the considered types defined in section 5.5 (point 2.2). A solution to some of the problems related to structural heterogeneity of attributes in descriptions mentioned in point 3 is presented in section 7.3. In chapter 8 we described a method to construct entity matching rules relying on bottom-up and top-down complementary approaches. Finally, in this chapter we need to define in details the process leading to the applications of entity matching rules to compute entity matching decision (point 5).

9.1 Computing String Similarity

In this section we briefly describe the practical problems related to the computation of an knowledge-based entity matching decision. Once the solution to semantic and structural heterogeneity are applied, we need to compute similarity score between the values of semantically equivalent features composing the compared descriptions. Then, given these similarity scores, we have to verify whether any of the matching rules defined as described in section 6 is satisfied.

It is clear that the similarity metric chosen affects the similarity score of attributes when these are represented with syntactic variations across different datasets. This problem was deeply studied in the past, and one single solution satisfying all of the cases was not found. However, it is clear that using a similarity metric ω_{eq} that returns similarity 1 if and only if the string are equal, would produce a similarity vector where each $w_i \in v$ is either 1 or 0. This similarity metric however would not allow to capture similar but not equal attributes. In section 3.3 we reviewed some similarity metrics that were defined in the past years to solve problems related to record linkage. In this context, we do not deal directly with the string matching problem, and we decided to simply use existing implementations of the most common similarity metrics. At this regards, we relied on SimMetrics 1.6.2 java library[1], and we decided to experimentally evaluate each of them to understand whether there is a similarity metric that is more suitable to be employed in an open syntactically heterogeneous context.

[1] SimMetrics is a Similarity Metric Library, e.g. from edit distance's (Levenshtein, Gotoh, Jaro etc) to other metrics, (e.g Soundex, Chapman). Work provided by UK Sheffield University funded by (AKT) an IRC sponsored by EPSRC, grant number GR/N15764/01. http://sourceforge.net/projects/simmetrics/

9.1.1 Best Similarity Metric Per Feature

Finding a single similarity metric that applies effectively to all the different features may be impossible. Therefore, by choosing a single similarity metric, we are aware that we are selecting the one that works better in average. This type of approach may be sub-optimal, as we could exploit our knowledge related to the semantic of the features to use the best similarity metric for each of the features types. There exists machine learning techniques aimed at feature extraction that we could employ to attempt guessing the best similarity metric for each of the considered types.

In section 8.2.2 of chapter 8 we described the labeling process of pairs of samples aimed the creation of a training set for the extraction of entity matching rules. We could exploit the same training set in order to apply feature extraction techniques and learn what is the best similarity metric that works for each of the considered features and entity type.

Among the existing one, we decided to rely on a feature extraction process based on Support Vector Machines (SVM) [49, 77]. SVM are binary classifiers, in which a dataset is represented as a highly dimensional space. Every feature characterizing the data represents a dimensions. All the dimensions are represented in a vector. Given a training set of samples labeled according to the objective of the classification, a Support Vector Machine looks for the largest margin based on the hyper-plans defining the borders of the classes. This margin is aimed at optimally separating the given samples. Every sample is represented as a vector, and every element of the vector represent the value of a specific dimensions. The largest margin between the two classes in a highly dimensional space is determined by SVM learning algorithms. Then, when a previously unseen sample has to be classified, support vector machine estimates its position in the highly dimensional space with respect to the margin. The classification decision is then taken according to the position of the sample. In order to maximize the reliability of the classification decision taken by a SVM, the margin between the hyperplanes separating the classes must be maximized. Given the margin defined by the support vector machine, it is possible to estimate the weights of features that is used to compute the functional margin of a sample (i.e. the distance from the decision hyper-plan). Intuitively, the weights of features reflect how these are relevant for classification purposes as their evaluation falls closer to the decision hyper-plan. In this context, we don't feel the need of further unfolding the details of the theory underlying Support Vector Machines, and we limit to exploit this features for our purposes. For more details about SVM, please refer to [83].

We briefly discussed how Support Vector Machines can effectively estimate the weights of feature with respect to the classification purpose. However, Support Vector Machines are known to optimally learn these weights for the most important features, loosing precision in the tail when training set is small and there is a high number of dimensions [152]. Therefore, we defined a feature selection process based on a variant of the recursive feature elimination process as described in [72].

The goal of the feature extraction process is to select the similarity metric that works better for classification purposes on the different types of features. Therefore, what we did was to build for each of the samples contained in the training set, a similarity vector containing the similarity between the features according to each of the similarity metrics considered (e.g *name_equal*, *name_jaro*, *name_levensthein, etc....* Practically, we compared all the features using all the similarity metrics known, and we produced a similarity vector as a training sample for a Support Vector Machine. A trained support vector machines, produces a list of weights for the features used in the classification. The recursive feature elimination algorithm then simply consists in computing the module of the weight, ranking them, removing the best feature, and recompute the classification. A formalization of the process in the following steps. Given F as empty list of ranked features, and R a list of all considered features.

1. Represent training set according to features in R;

2. Train a support vector machine with the training set;

3. Remove the feature with the highest weight from R, and put in F

4. Repeat 1, 2 and 3 until R is empty;

At the end of the process, we have in F a ranked list of all features defined in the identification ontology tied with each of the similarity metric considered. By processing this list top-down, we can extract the best similarity metric for each of the feature. This process is just one among the possible feature extraction process. As the main goal of the thesis is not to solve this problem, we did not investigate deeply other existing solutions. However, Support Vector Machine and Recursive Feature Extraction are known to be effective classification and feature extraction techniques. Therefore, even thou we cannot claim to having defined the absolutely optimal solution, we are quite confident that proposed technique is fair enough for evaluation purposes. For this reason, we implemented the process described above in software component relying on Weka 3.6.5 [106], Apache Maven and Oracle Java 1.6. In particular, we relied on Platt's the Sequential Minimal Optimization (SMO) algorithm described in [120] and improved in [91]. Each training step was executed relying on a 10-fold cross validation, in order to reduce the bias of the learned weights.

9.2 Computing Features Comparison

Assuming we selected one similarity metric $\omega \in O$ that works better than the others, we now have another problem related to how to compare multiple instances of semantically equivalent features. In fact, as shown in section 5.5, there exists features that are not functional, and furthermore, there could also be several syntactic variations of the same feature. See for example the names:

name: Antônio Carlos Brasileiro de Almeida Jobim **name**: Jobim, Antonio Carlos

are part of the same description. Therefore, a question comes: which one should we compare? In principle, this question cannot have an absolutely correct answer a priori. In fact, any choice is prone to error. One would say that the best option is to compare the most complete rendering of the attribute, but this exposes the comparison to inconsistencies as shown in section 2.1. Consider for example the attributes:

`foaf:givenName`: *Antônio Carlos Brasileiro de Almeida Jobim*;
`foaf:givenName`: *Antônio Carlos*

In this case, the choice of the most complete (or long) version of the attribute would lead to error in the matching. Therefore, we believe that in principle, all the instances of equivalent features should be compared among each other. However, at the bottom of any comparison, we still have to decide whether a rule is satisfied or not. Therefore, we have to define a way to produce a unique similarity score representative of all the variations. In the following we propose three possible variants, with pros and cons.

9.2.1 Greedy Features Comparison

The most simple and straightforward approach is to greedily pick the best comparison result as the one representative of the similarity between two sets of semantically equivalent features. Namely, when comparing two sets of features, we select the max similarity score. In a sense, we have to define a greedy κ comparator defined in equation (6.5) at page 102. Formally, let's define $\kappa_{GREEDY} : \Omega \times \mathcal{S}[v_1, ..., v_n] \times \mathcal{S}[v_1, ..., v_m] \rightarrow \Re \in [0, 1]$ as the function implementing the following process:

```
greedyComparator(List features1, List features2, Comparator o){
   max = 0.0;
   for each f1 in features1{
      for each f2 in features2{
         score = o.compare(f1,f2);
         if(score==1.0){
            return score;
         }else if(score>max){
            max = score;
 }
      }
   }
   return max;
}
```

The process described above guarantees a maximization of the similarity score between equivalent features, neglecting incompatible syntactical variations. However, this approach suffers of issues related to incompleteness of some attribute variations, as the best match would greedily choose the best match, regardless the completeness of the attribute. To better clarify this aspect, consider the examples presented in table 9.1. Descriptions 1 and 2 are about two Brazilian football players. They have in principle very different names, but they both are known by the name they put on their t-shirt while playing football. Besides the fact that the descriptions do not match, by choosing a greedy approach in matching this feature, we'd have a false perfect matching score.

1	**name**: Emerson Moises Costa **name**: Emerson Moisés Costa **name**: Emerson **birthdate**: 1972-04-12 ...
2	**name**: Émerson Ferreira da Rosa **name**: Rosa, Emerson Ferreira da **name**: Emerson **birthdate**: 1976-04-04

Table 9.1: Examples problematic greedy match of features

9.2.2 Features Comparison with Relative Completeness

In order to reduce the effects of the adoption of a greedy approach, we introduce now the concept of *Relative Completeness* as measure of the completeness of a value of a feature with respect to other values of the same feature in the same description. The completeness of a feature measures how the value of a feature is complete with respect to its most complete rendering. To have a measure of completeness about any feature would require to know the most complete value of any feature a priori. As this is not possible in principle, we limit our analysis of completeness relatively to the description where the feature appears. Thereby, what we can consider in this context is simply the *Relative Completeness* of the feature. In words, the estimation of the relative completeness of a feature consists in estimating how complete is the value of a feature by computing a ratio with respect to the most complete value in the same description. In this context, we assume that the most complete value is the longest one in the description. Thus, given c_i as a measurement of syntactical length of the value of a feature, and c_{max} as the length of the longest feature value among the semantically equivalent one in the descriptions, we can compute $RC_i = \frac{c_i}{c_{max}}$. To give a practical example, given the three values of the attribute name contained in the description 2 of table 9.1:

name: Émerson Ferreira da Rosa

name: Rosa, Emerson Ferreira da

name: Emerson

considering a *token based relative completeness*, the firsts two attributes would have relative completeness of $RC_1 = \frac{4}{4} = 1$, whereas the third attribute would have a relative completeness $RC_3 = \frac{1}{4} = 0.25$. The relative completeness estimated in this way is then used to weight the similarity score obtained by the comparison. Therefore, if we compare *Emerson* with *Emerson*, the similarity score will be weighted to be affected by the low weight, and we would reduce the problem of false positive matching. Let's define κ_{RC}, as the function implementing the following process:

```
greedyRCComparator(List features1, List features2, Comparator o){
   max = 0.0;
   relative-completeness-max-1 = 0.0;
   relative-completeness-max-2 = 0.0;

   for each f1 in features1{
      if(f1.length > relative-completeness-max-1)
          relative-completeness-max-1 = f1.length;
   }

   for each f2 in features1{
      if(f2.length > relative-completeness-max-2)
          relative-completeness-max-1 = f2.length;
   }

   for each f1 in features1{
      for each f2 in features2{
         rc1 = f1.length / relative-completeness-max-1;
         rc2 = f2.length / relative-completeness-max-2;
         score = o.compare(f1,f2);
         normalizedScore = rc1 * rc2 * score;
         if(normalizedScore==1.0){
            return normalizedScore;
         }else if(normalizedScore>max){
            max = normalizedScore;
 }
      }
   }
   return max;
}
```

Notice that in principle we can estimate the completeness of a feature considering different level of granularity. For example, we can consider *character-based* completeness estimation (like the one in the pseudo-code above), or *token-based* completeness estimation relying on the number of different words composing the feature values. Both approaches have pros and cons that will be evaluated experimentally.

9.2.3 Features Comparison considering Average Score

The application of *Relative Completeness* may reduce the number of false positive matching when comparing different incomplete syntactic variations of the same at-

1	**name**: Rovereto **contained by**: Provincia di Trento; **contained by**: Trentino Alto Adige; **contained by**: Italy; **contained by**: Europe; ...
2	**name**: Rovereto **contained by**: Provincia di Modena; **contained by**: Emilia Romagna; **contained by**: Italy; **contained by**: Europe; ...

Table 9.2: Examples problematic greedy match of non-functional features

tribute values. However, this might not always be the case. In fact, all the non-functional attributes may present legally different values for the same feature. Therefore, applying a greedy approach in this context may produce further distortions. For examples, consider the property *contained by* defined for the entity type Location. Consider the example of Rovereto, as a location in Trentino and Emilia Romagna presented in table 9.2. Is pretty clear that the locations do not match because they are contained in different locations, and this information is explicitly represented in the description. However, a greed approach in matching the *name* and the *contained by* features would lead to a false positive match. In fact, the perfect match of “Italy” as value of the feature contained by would produce a 1.0 as similarity between the two features. To avoid this type of inconvenience we propose to consider the average best score of all the instances of the features. Let’s define κ_{AVG} as the function implementing the following process:

```
averageComparator(List features1, List features2, Comparator o){
   if(feature2.size < feature1.size){
      tmp = feature2;
      feature2 = feature1
      feature1 = tmp;
   }
   maxSum = 0.0
   for each f1 in features1{
     max = 0.0;
      for each f2 in features2{
          score = o.compare(f1,f2);
          if(score>max){
            max = score;
  }
      }
      maxSum=maxSum + max;
   }

   return maxSum/features1.size;
}
```

It is important to notice the average is computed on the smallest number of features, and if one of the list contains just one feature, then the matching process is equivalent to the greedy one.

9.3 Computing Rule-Based Matching Decision

So far, we described how we can deal with the comparison of multiple instances of the same feature type (a.k.a semantically equivalent features). The management of this process allows us to produce a similarity score between features contained in different descriptions. Now we need to apply the result of such process to verify the satisfaction of rules for entity matching.

A simple way to compare descriptions is to greedily attempt to apply all the matching rules until one is satisfied. Every rule explicitly supports a matching decision, and when satisfied a matching decision can be taken. In case no rule is satisfied, we assume that the final decision is *Don't Know*.

Intuitively, the order of application of rules may affect the final matching decision. In fact, as analyzed in chapter 2 there may be inconsistencies or errors in the data contained in the descriptions. Therefore, we cannot exclude the case where pairs of descriptions satisfy positive and negative matching rules at the same time. Therefore, we have to careful in deciding both the order and the nature of the rule application process. If we apply first positive matching rules, we may have some false positive more, whereas if we apply first negative matching rules, we are likely to have some false negative. This for example the cases where fiscal code of two person match, but the date of birth is different. This is a clear inconsistency in the data, but how do we choose which one wins? Our intuition is that the satisfaction of positive matching rules is harder in principle than the satisfaction of negative matching rules. In fact, whenever the syntactic structural harmonization of attributes such as a date does not work, we may end up comparing information that are equal, but syntactically very different (e.g. “12/02/1982” and “12 of February 82”). As we cannot assume that the compared descriptions rely on homogeneous syntactic representations, we should give more value to matching instances of the attributes rather than the one that do not match. Therefore, we believe that positive matching rules should in general be applied first, in a greedy context.

9.4 Fingerprint Similarity

In the previous sections we outlined solutions to some of the issues related to the computation of knowledge-driven rule-based entity matching. In particular, we presented solutions related to the problem of the selection of the similarity metric to adopt, we analyzed pros and cons of feature comparison techniques, and we briefly discussed approaches for the application of rules.

In this section, we want to describe how we can exploit partially satisfied rules to define a simple distance-based solution that can be applied whenever the complete satisfaction of a rule is not possible. In fact, depending on how restrictive the rules are, we may end up in comparing examples where not positive or negative rule is satisfied.

In this case, we want to draw a $DONTKNOW$ matching decision, but we also want to estimate how similar the compared descriptions are.

In section 9.1.1, we mentioned how trained Support Vector Machine (SVM) classifiers can produce weights for the features according to their relevance in supporting classification choices. Notice that in this context, we do not deal with the technical details related to SVM, but we simply use them exploiting their known effectiveness for classification and regression purposes. Let's assume that relying on a trained support vector machine, we can gather a weight estimation for each of the features defined in the identification ontology outlined in chapter 5. Define this $W : w_1, ..., w_n$ as the list of weight for features, where w_i is weight of the i-th feature.

We now want to estimate the distance between two description whenever it was not possible to satisfy any of the defined rules. When this happens, we want to backtrack, and select the best partially satisfied rule to compute a distance between the compared descriptions. We defined as the best partially satisfied rule as the positive matching rule with the highest sum of weights of satisfied atoms, among the one with the highest ratio of satisfied atoms. More formally, let's define as SR the *satisfaction ratio of a rule*, as the ratio between the atoms satisfied in a rule and the atoms composing the rule ρ. $SR : \frac{|satisfy_\theta \in \rho|}{|\theta \in \rho|}$. Intuitively, the higher the number of satisfied atoms, the higher the ratio. A completely satisfied rule has $SR = 1$. Satisfaction ratio of rules may not be enough to discriminate the best partially satisfied rule, in fact there may be rules presenting the same number of atoms, and partially satisfied by different set of features compared. Therefore, among partially satisfied rules with equal *satisfaction ratio*, we want to select the one minimum weight the feature embedded in the unsatisfied atoms. Hence, let's define the function $\delta_{SAT\rho}$ as the function that extract the feature of satisfied atoms:

$$\delta_{SAT\rho}(\rho) : \{\alpha \in A | \exists \theta \in \rho \wedge \alpha \in \theta \wedge satisfy_\theta(\theta)\} . \tag{9.1}$$

Then, relying on the $\delta\rho$ function extracting the features used in a rule as defined in equation (6.2) at page 101, we can extract the names of the unsatisfied atoms of a rule ρ simply by computing $\delta_\rho(\rho) \setminus \delta_{SAT\rho}(\rho)$. Given all the rules with equal *satisfaction ratio*, we can select the one with minimum unsatisfied weight. If unsatisfied atoms happen to have the same weight, then we can randomly select one of those.

Now that we defined how to select the best partially satisfied rule, we can proceed computing the similarity distance between unknown matching pairs. Given the list of weights of the feature of the partially satisfied rule $[w_1, ..., w_m]$, with m as the number of token composing the rule ρ, and the similarity scores computed for the satisfaction of n atoms $[sim_1, ..., sim_n]$ of the rule ρ using a string similarity metric $\omega \in \Omega$, then we compute the *fingerprint similarity* as follows:

$$fp(\rho, [sim_1, ..., sim_n], [w_1, ..., w_m]) : \frac{\sum_{i=0}^{n} w_i \cdot sim_i}{\sum_{i=0}^{m} w_i} \tag{9.2}$$

Relying on this formula, we can now compute a similarity score that can be used for ranking purposes when no sharp rule can be reliably satisfied. In the next chapter, we will evaluate the effectiveness of this similarity metric, and compare it with the Feature-Based Entity Matching algorithm [139], currently deployed as default matching module in the Okkam Entity Name System.

Chapter 10

Experimental Evaluation

In the third part of the thesis, we described the tools aimed to be part a knowledge-based solution for open entity matching. In chapter 7 we presented simple and straightforward solutions to the problems of semantic and structural heterogeneity. In particular, we present a part of the contextual mappings defined, and how we exploited the semantic of some attributes to reduce structural heterogeneity due to different granularities in representation of features like name, date, geo-coordinates.

The pursue for a solution to the entity matching in the context of the (Semantic) Web is motivated also by the unreliability of large part of *owl:sameAs* statements [73, 74]. We believe that spurious *owl:sameAs* statements undermine the development of the Semantic Web. Therefore, we decided to test the Fingerprint Match solution on the following dataset:

- evaluation on manually annotated dataset (person, location, organization);
- evaluation based on New York Times datasets (person, location organization);
- evaluation based on OAEI Instance Matching 2010[1] (person, organization);

These will be described more in details in section 10.1.

The evaluation of any possible combination of factors affecting the construction of rules described in this thesis is not feasible in the context of this work. In particular we decided to evaluate the factors presented in the following paragraphs.

At the bottom of any matching solution lies a string similarity metric. This factor surely affects both the quality of the learning process as well as the application of rules. Furthermore, we want to evaluate also the impact of different methods of comparison. In particular we choose to evaluate and compare the Simple Greedy approach, the Knowledge-driven approach and the Greedy approach with Relative Completeness

[1] `http://oaei.ontologymatching.org/`

Character-based. We choose these three because they are representative all the variations presented above. Namely, we consider simple greedy approach, a combination of greedy and average approach, and a greedy approach relying on relative weight estimation to normalize similarity scores.

Another important aspect to evaluate, reminding that one of hypothesis to test that mixing bottom-up and top-down rules can provide benefits with respect to adoption of these approaches alone, is the impact of the rules inconsistency normalization process. That is, we want to evaluate whether the definition of shorter rules applying inconsistency removal, and the definition of more conservative rules applying inconsistency atom normalization have some impact on the overall matching process.

Besides atom inconsistency normalization, an important aspect to consider is how we normalize the string similarity thresholds for the satisfaction of rules atoms in both positive and negative matching rules. In fact, the selection of conservative thresholds should result in a more conservative classification, with a larger set of samples classified as *don't known*. Conversely, the adoption of relaxed thresholds may negatively affect the precision of the classification.

Another important factor affecting the generation of rules for open entity matching, is the process of integration of top-down and bottom-up learned rules. In this context we are interested in evaluating the impact of a plain integration of all top-down rules, compared with the integrations of only positive and only negative top-down defined matching rules.

The evaluation of the combination of all these factors for all the datasets and the considered entity types would produce already a quite extensive set of experiments to run. However, given the theoretical foundation given in chapter 6.1 for the definition of rules, we want also to evaluate the impact of a binary versus a three-class classification. In fact, it seems interesting to evaluate how the presence of the *unknown* labeled samples affects the learning rules process.

In section 10.2.1 we present an evaluation of proposed method relying only top-down extracted rules. In section 10.2.2 we propose an evaluation of the methods to learn rules from a training set. Finally, in section 10.2 we evaluate the integrated solution mixing top-down and learned rules.

10.1 Evaluation Datasets

In this section we aim at describing the evaluation datasets, providing descriptive statistics as the one defined for training sets presented in sections 8.2.2, 8.2.2, and 8.2.2. In order to make the description of the datasets more compact, we choose tabular form

Evaluation Set Data				Sample Distribution																				
				DBP			Fac			FrB			LFM			MBz			Okk			OCs		
Source	$\|D\|$	$\lambda(A)$	$\sigma(A)$	M	N	U	M	N	U	M	N	U	M	N	U	M	N	U	M	N	U	M	N	U
DBP	208	84.8	39.7	0	89	0	7	102	4	87	94	1	14	80	2	10	206	18	22	465	1	3	152	4
Fac	76	21.5	5.0	-	-	-	2	1	1	0	73	1	0	25	1	0	103	2	1	12	1	1	3	0
FrB	99	59.0	43.9	-	-	-	-	-	-	1	32	1	4	66	3	6	229	14	13	22	4	1	0	0
LFM	54	31.7	17.07	-	-	-	-	-	-	-	-	-	0	1	0	9	76	33	1	4	1	0	0	0
MBz	82	9.4	5.6	-	-	-	-	-	-	-	-	-	-	-	-	0	16	0	0	9	1	0	0	0
Okk	75	11.9	1.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	10	0	0	4	0
OCs	14	16.3	6.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0

Table 10.1: Manually Annotated Evaluation Set Description for Person.

this time. The columns named $|D|$ indicate the number of description from a source. The columns named with $\lambda(A)$ indicate the average number of attributes for descriptions, whereas $\sigma(A)$ indicates the standard deviation. Furthermore, we compressed the names in acronyms to further compact the tables. In particular *DBP* stands for DBPedia, *Fac* stands for Factual, *FrB* stands for Freebase, *LFM* stands for LastFM, *MBz* stands for MusicBrainz, *Ok* stands for Okkam, *OCs* stands for OpenCongress, *Geo* stands for Geonames, *OCPs* stands for OpenCorporates.

10.1.1 Person Evaluation Datasets

The datasets used to evaluate rules extracted for entity type person are:

1. A manually labeled evaluation set, result of the same process used for the generation of the training set, but starting from a different set of seed queries. This evaluation set is described in table 10.1, and contains 2145 samples (182 positive samples, 1874 negative samples, and 89 unknown), involving 608 different descriptions and 28752 different attributes of 1689 different types.

2. The New York Times dataset for people[2], aiming at discovering the *owl:sameAs* between Freebase and DBpedia. The dataset is formed by 4614 positive matching pairs, with $\lambda(A) = 101.53$ and $\sigma(A) = 52.42$ for DBpedia descriptions, and $\lambda(A) = 109.29$ $\sigma(A) = 77.91$ for freebase descriptions. Notice that DBPedia and Freebase descriptions present 468192 attributes of 1709 different types and 508969 attributes of 1414 different types respectively.

3. A dataset used from OAEI Instance Matching evaluation 2010, composed by 900 positive matching pairs for person descriptions containing different types of perturbations. In particular the dataset contains 2000 different descriptions with $\lambda(A) = 13$ and $\sigma(A) = 0$. Therefore, the total amount is 26000 attributes of 19 different types. Notice that the descriptions matched were constructed traversing the RDF graph provided. For example, OAEI person dataset presented street ad-

[2] `http://data.nytimes.com/people.rdf`

Evaluation Set Data				Sample Distribution														
				DBP			Geo			Fac			FrB			Okk		
Source	$\|D\|$	$\lambda(A)$	$\sigma(A)$	M	N	U	M	N	U	M	N	U	M	N	U	M	N	U
DBP	28	100.8	55.7	0	12	0	3	33	0	0	20	0	0	1	0	0	68	0
Geo	78	13.01	0.4	-	-	-	0	45	0	2	43	2	0	0	0	9	75	0
Fac	29	16.75	3.51	-	-	-	-	-	-	1	29	1	0	0	0	2	34	0
FrB	1	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0
Okk	45	12.42	1.8	-	-	-	-	-	-	-	-	-	-	-	-	1	30	3

Table 10.2: Manually Annotated Evaluation Set Description for Location.

dress of people as resources. Therefore, to compose an address we had to traverse the graph until literal values were found.

10.1.2 Location Evaluation Datasets

The datasets used to evaluate rules extracted for entity type location are:

1. A manually labeled evaluation set, result of the same process used for the generation of the training set, but starting from a different set of seed queries. This evaluation set is described in table 10.2, and contains 414 samples (18 positive samples, 390 negative samples, and 6 unknown), involving 181 different descriptions and 5336 different attributes of 430 different types.

2. The New York Times dataset for people[3], aiming at discovering the *owl:sameAs* between Freebase. Geonames and DBpedia. The dataset is formed by 3577 positive matching pairs, with as described in table 10.3. The evaluation set contains 5452 different descriptions presenting 665478 attributes of 3809 different types.

Evaluation Set Data				Sample Distribution								
				DBP			Geo			FrB		
Source	$\|D\|$	$\lambda(A)$	$\sigma(A)$	M	N	U	M	N	U	M	N	U
DBP	1826	136.6	77.3	37	0	0	1127	0	0	1190	0	0
Geo	1749	77.7	85.8	-	-	-	0	0	0	1122	0	0
FrB	1877	149.1	133.4	-	-	-	-	-	-	0	0	0

Table 10.3: New York Times Evaluation Set Description for Location.

10.1.3 Organization Evaluation Datasets

The datasets used to evaluate rules extracted for entity type person are:

1. A manually labeled evaluation set, result of the same process used for the generation of the training set, but starting from a different set of seed queries. This evaluation set is described in table 10.4, and contains 1511 samples (45 positive samples, 1430 negative samples, and 27 unknown), involving 812 different descriptions presenting 20062 different attributes of 920 different types.

[3] `http://data.nytimes.com/location.rdf`

2. The New York Times dataset for people[4], aiming at discovering the *owl:sameAs* between Freebase and DBpedia. The dataset is formed by 4614 positive matching pairs, with $\lambda(A) = 72.08$ and $\sigma(A) = 33.83$ for DBpedia descriptions, and $\lambda(A) = 114.15$ $\sigma(A) = 68.37$ for freebase descriptions. Notice that the 1872 descriptions from DBPedia presented 134951 attributes of 1601 different types, and the 1873 descriptions from Freebase presented 213817 attributes of 1711 different types. respectively.

3. A dataset used from OAEI Instance Matching evaluation 2010, composed by 89 positive matching pairs for real world restaurant descriptions containing different types of perturbations. In particular the dataset contains 754 different descriptions with $\lambda(A) = 7$ and $\sigma(A) = 0$. Therefore, the total amount is 5278 attributes of 9 different types. Notice that also in this case, the matched descriptions are constructed by traversing the RDF provided. For example, OAEI restaurant dataset presented street addresses of restaurants as resources. Therefore, to compose an address we had to traverse the graph until literal values were found.

10.2 Evaluating Fingerprint Match

In this section we evaluate the performances of the knowledge based solution we defined. To do so, we are going to run a set of experiments and estimate accuracy relying on standard metrics such as precision, recall and f-measure. Precision ($\frac{\#TP}{\#TP+FP}$), recall ($\frac{TP}{TP+FN}$) and F-measure ($2 \times \frac{precision \times recall}{precision+recall}$), according to standard record linkage evaluation methods. Furthermore, we want to estimate a more custom accuracy metric we named $\rho - accuracy$, that allows to compute a general accuracy measure reflecting the principle of our evaluation. Namely, we would like to evaluate methods, not only based on how well it does on discovering matching pairs, but also how conservatively it takes negative matching decisions. In order to do so, we compute a different evaluation measure reducing the effect of conservative don't know decisions, and penalizing greedy

[4] http://data.nytimes.com/organization.rdf

Evaluation Set Data				Sample Distribution																	
				DBP			Fac			FrB			MBz			Okk			OCPs		
Source	$\|D\|$	$\lambda(A)$	$\sigma(A)$	M	N	U	M	N	U	M	N	U	M	N	U	M	N	U	M	N	U
DBP	88	73.8	39.5	0	65	1	5	138	2	19	75	1	0	11	0	5	18	0	0	38	0
Fac	351	16.8	7.0	-	-	-	1	171	0	4	162	6	0	28	1	0	0	0	0	108	2
FrB	136	31.0	36.6	-	-	-	-	-	-	2	151	7	5	14	1	0	0	0	4	139	0
MBz	7	7.7	1.7	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
Okk	30	10.7	1.2	-	-	-	-	-	-	-	-	-	-	-	-	0	15	0	0	12	0
OCPs	191	15.7	4.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	258	6

Table 10.4: Manually Annotated Evaluation Set Description for Organization.

false positive match classifications. More formally:

$$\rho - accuracy = \frac{TP + TN + TD}{(k \times FP) + FN + (\frac{1}{k} \times FD) + TP + TN + TD} \tag{10.1}$$

With TP as *True Positive Rate*, TN as *True Negative Rate*, TD as *True DontKnow Rate* FD as *False DontKnow Rate*, FP as *False Positive Rate* and FN *False Negative Rate*, and $K > 1$. This way we aim giving a higher score to the experiment configuration that discover a higher number of positive match, and penalizes the false positives. Notice that we price False DontKnow classification as we rather have a conservative matcher aiming at defining a reliable matching decision. For our experiments, we relied on a ρ-accuracy with $k = 2$.

10.2.1 Top-down Only Rules Experiments

In this section we aim at evaluating our matching solution relying on matching rules extracted only relying on the terms defined in the identification ontology. Due to the low number of positive matching attributes (i.e. inverse-functional properties), we decided to define positive and negative matching rules also relying on combinations of functional properties. Given the large number of functional properties included in the ontology, we had to limit the size of combination of attributes considered. Therefore, relying on the assumption that the matching of the *name* is an essential attribute, then we would also consider as a *match any combination of attributes of 3 functional attributes plus the name* with a similarity threshold of at least 0.9, and we would also consider *as non-matching all functional attributes plus the name* with a similarity threshold below 0.5. Then we evaluated the rule generation process relying on 3 different matching comparison approaches, and considering all the similarity metrics singularly. We are aware that the choice of the number of attributes is arbitrary, as well as the threshold, but the evaluation of this type of approach is out of the scope of this work. In fact, discovering the combinatorial generation of rule is not the goal of this thesis, and there exists very sophisticated regression methods as the one described in [118, 52, 88]. We believe that 0.9 is a fair threshold to consider an attribute as matching, as it was used also in other works such as [118]. Furthermore, also 0.5 seems to be is a fair threshold for non matching attributes. All cases not satisfying any of these clauses had to be considered unknown.

Top-down Rules for Person

In table 10.5 we present the results of matching experiment executed relying top-down defined rules. The matching algorithms applying the rules produced quite impressively

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.929	0.929	0.929	0.929	0.929	0.817	0.939	0.929	0.938	0.929	0.922
	Recall	0.298	0.298	0.298	0.298	0.298	0.443	0.351	0.298	0.344	0.298	0.359
	F-Measure	0.451	0.451	0.451	0.451	0.451	0.574	0.511	0.451	0.503	0.451	0.516
NonMatch	Precision	0.895	0.939	0.939	0.94	0.939	0.962	0.955	0.939	0.957	0.939	0.948
	Recall	0.729	0.977	0.977	0.977	0.977	0.892	0.836	0.977	0.814	0.977	0.977
	F-Measure	0.804	0.958	0.958	0.958	0.958	0.926	0.892	0.958	0.88	0.958	0.962
DontKnow	Precision	0.031	0.252	0.252	0.26	0.252	0.186	0.139	0.252	0.143	0.252	0.277
	Recall	0.182	0.295	0.295	0.307	0.295	0.591	0.625	0.295	0.716	0.295	0.352
	F-Measure	0.053	0.272	0.272	0.281	0.272	0.283	0.228	0.272	0.238	0.272	0.31
Weighted Avg	Precision	0.859	0.909	0.909	0.909	0.909	0.919	0.918	0.909	0.921	0.909	0.917
	Recall	0.678	0.904	0.904	0.905	0.904	0.85	0.796	0.904	0.78	0.904	0.91
	F-Measure	0.748	0.896	0.896	0.896	0.896	0.875	0.838	0.896	0.828	0.896	**0.905**
	ρ-Accuracy	0.771	0.921	0.921	**0.922**	0.921	0.896	0.869	0.921	0.86	0.921	**0.927**

Table 10.5: Greedy Comparison on Manually Annotated Dataset for Person

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.929	0.929	0.929	0.929	0.929	0.934	0.939	0.929	0.939	0.929	0.922
	Recall	0.298	0.298	0.298	0.298	0.298	0.435	0.351	0.298	0.351	0.298	0.359
	F-Measure	0.451	0.451	0.451	0.451	0.451	0.594	0.511	0.451	0.511	0.451	0.516
NonMatch	Precision	0.895	0.939	0.939	0.94	0.939	0.962	0.955	0.939	0.958	0.939	0.948
	Recall	0.729	0.977	0.977	0.977	0.977	0.896	0.836	0.977	0.814	0.977	0.977
	F-Measure	0.804	0.958	0.958	0.958	0.958	0.928	0.892	0.958	0.88	0.958	0.962
DontKnow	Precision	0.031	0.252	0.252	0.26	0.252	0.187	0.139	0.252	0.143	0.252	0.277
	Recall	0.182	0.295	0.295	0.307	0.295	0.602	0.625	0.295	0.716	0.295	0.352
	F-Measure	0.053	0.272	0.272	0.281	0.272	0.286	0.228	0.272	0.238	0.272	0.31
Weighted Avg	Precision	0.859	0.909	0.909	0.909	0.909	0.927	0.918	0.909	0.921	0.909	0.917
	Recall	0.678	0.904	0.904	0.905	0.904	0.853	0.796	0.904	0.78	0.904	0.91
	F-Measure	0.748	0.896	0.896	0.896	0.896	0.879	0.838	0.896	0.829	0.896	0.905
	ρ-Accuracy	0.771	0.921	0.921	**0.922**	0.921	0.903	0.869	0.921	0.86	0.921	**0.927**

Table 10.6: Knowledge-driven Comparison on Manually Annotated Dataset for Person

good results. In particular, the similarity metrics performing slightly better than the others is Taglink, but the others also performed well. The non matching decisions were taken with a high precision, but not perfectly. This may be due to the fact that the number of don't know labeled samples is relatively high, causing somehow troubles in taking non matching decisions. Positive matching decisions were taken with a high level of precision, but not perfectly as well. The presence of don't know labeled samples may explain also in this case a non perfect precision. It is interesting to notice how the ρ-accuracy has a higher score than the weighted F-measure. This is probably due to the fact that many ambiguous cases both from match and non match were classified as don't know. This may be the effect of the structural heterogeneity among the compared descriptions, decreasing the number of rules satisfied with at least 4 shared matching attributes. This fact is also reflected by the low precision of don't know sample classification. Knowledge-driven comparison method produces the same results as shown in table 10.6. Slightly worst performances were the result of character-based relative completeness estimation method, which apparently affected the recall of positive matches without improving the precision only in few cases as expected.

The experiment on the NYT dataset using top-down rules generated with combining functional properties defined in the identification ontology produced quite interestingly

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.929	0.927	0.938	0.923	0.939	0.821	0.938	0.941	0.927	0.938	0.927
	Recall	0.298	0.29	0.229	0.275	0.237	0.42	0.344	0.244	0.29	0.229	0.29
	F-Measure	0.451	0.442	0.368	0.424	0.378	0.556	0.503	0.388	0.442	0.368	0.442
NonMatch	Precision	0.895	0.937	0.931	0.936	0.93	0.958	0.954	0.932	0.95	0.93	0.939
	Recall	0.729	0.977	0.977	0.977	0.977	0.896	0.836	0.977	0.818	0.977	0.977
	F-Measure	0.804	0.957	0.954	0.956	0.953	0.926	0.891	0.954	0.879	0.953	0.957
DontKnow	Precision	0.031	0.242	0.25	0.26	0.255	0.189	0.14	0.25	0.144	0.253	0.279
	Recall	0.182	0.273	0.273	0.295	0.273	0.58	0.625	0.273	0.705	0.273	0.33
	F-Measure	0.053	0.257	0.261	0.277	0.264	0.285	0.228	0.261	0.239	0.262	0.302
Weighted Avg	Precision	0.859	0.906	0.902	0.906	0.902	0.916	0.917	0.903	0.914	0.901	0.909
	Recall	0.678	0.903	0.899	0.903	0.899	0.852	0.795	0.9	0.779	0.899	0.905
	F-Measure	0.748	0.893	0.886	0.893	0.886	0.874	0.837	0.888	0.823	0.886	0.896
	ρ-Accuracy	0.771	0.919	0.914	0.92	0.915	0.894	0.867	0.915	0.857	0.914	0.921

Table 10.7: Character-based RC weighted Comparison on Manually Annotated Dataset for Person

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Greedy	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.512	0.524	0.514	0.514	0.591	0.731	0.609	0.513	0.552	0.527	0.779
	F-Measure	0.677	0.687	0.679	0.679	0.743	**0.844**	0.757	0.678	0.711	0.691	**0.876**
	ρ-Accuracy	0.514	0.536	0.526	0.526	0.604	**0.761**	0.628	0.525	0.572	0.539	**0.802**
Knowledge	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.512	0.524	0.514	0.514	0.591	0.735	0.609	0.513	0.552	0.527	0.779
	F-Measure	0.677	0.687	0.679	0.679	0.743	0.848	0.757	0.678	0.711	0.691	0.876
	ρ-Accuracy	0.514	0.536	0.526	0.526	0.604	**0.765**	0.628	0.525	0.572	0.539	**0.802**
Match	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.512	0.472	0.416	0.478	0.507	0.683	0.581	0.421	0.441	0.39	0.695
	F-Measure	0.677	0.642	0.587	0.646	0.673	0.812	0.735	0.592	0.612	0.561	0.82
	ρ-Accuracy	0.514	0.483	0.425	0.488	0.517	0.712	0.599	0.43	0.457	0.4	0.713

Table 10.8: Matching Experiment on NYT Dataset for Person

a good number of positive matching decisions (table 10.8). The fact that also matching with string similarity produced a relatively good results is a sign that the transformation function applied to normalize attributes names, and dataset, positively affected matching performances. However, best similarity metrics for matching entity types persons of the NYT dataset are Taglink and Monge-Elkan, followed by Overlap and Jaro-Winkler. The others performed closely to equal. It is important to highlight that the ρ-accuracy is lower than the F-measure standard. This shows that several positive matching samples were classified as non matching rather than don't know. However, the difference between F-measure and ρ-accuracy is lower for Taglink, showing to perform more reliably. The comparison methods produced similar results, with a slight improvement of performances using Monge-Elkan and Knowledge-driven comparison process.

In table 10.9, we present the result of the matching experiments with top down defined rules on the OAEI 2010 dataset. As shown in the table, all similarity metrics performed very well. There is not difference among the comparison methods, because the data do not present variations, and the matching of the attributes present different types of syntactical perturbations. Furthermore, all samples present the same 13 semantically equivalent attributes, granting the satisfaction of the rules. The similarity

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Greedy	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.96	0.968	0.961	0.961	0.991	0.966	0.961	0.96	0.963	0.977	0.968
	F-Measure	0.98	0.984	0.98	0.98	**0.996**	0.982	0.98	0.98	0.981	0.988	0.984
ρ-Accuracy	0.96	0.968	0.961	0.961	**0.991**	0.966	0.961	0.96	0.963	0.977	0.968	
Knowledge	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.96	0.968	0.961	0.961	0.991	0.966	0.961	0.96	0.963	0.977	0.968
	F-Measure	0.98	0.984	0.98	0.98	**0.996**	0.982	0.98	0.98	0.981	0.988	0.984
	ρ-Accuracy	0.96	0.968	0.961	0.961	**0.991**	0.966	0.961	0.96	0.963	0.977	0.968
Match	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.96	0.968	0.961	0.961	0.991	0.966	0.961	0.96	0.963	0.977	0.968
	F-Measure	0.98	0.984	0.98	0.98	**0.996**	0.982	0.98	0.98	0.981	0.988	0.984
	ρ-Accuracy	0.96	0.968	0.961	0.961	**0.991**	0.966	0.961	0.96	0.963	0.977	0.968

Table 10.9: Greedy Comparison OAEI dataset for Person

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	1.0	1.0	1.0	1.0	1.0	0.333	1.0	1.0	1.0	1.0	1.0
	Recall	0.056	0.056	0.056	0.056	0.056	0.111	0.056	0.056	0.056	0.056	0.056
	F-Measure	0.105	0.105	0.105	0.105	0.105	0.167	0.105	0.105	0.105	0.105	0.105
NonMatch	Precision	0.929	0.944	0.944	0.944	0.944	0.946	0.944	0.944	0.946	0.944	0.944
	Recall	0.033	1.0	1.0	1.0	1.0	0.987	1.0	1.0	0.997	1.0	1.0
	F-Measure	0.065	0.971	0.971	0.971	0.971	0.966	0.971	0.971	0.971	0.971	0.971
DontKnow	Precision	0.015	NaN	NaN	NaN	NaN	0.0	NaN	NaN	0.0	NaN	NaN
	Recall	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	F-Measure	0.03	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Weighted Avg	Precision	0.918	NaN	NaN	NaN	NaN	0.905	NaN	NaN	0.935	NaN	NaN
	Recall	0.048	0.944	0.944	0.944	0.944	0.935	0.944	0.944	0.942	0.944	0.944
	F-Measure	0.066	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	ρ-Accuracy	0.092	0.944	0.944	0.944	0.944	0.927	0.944	0.944	0.944	0.944	0.944

Table 10.10: Greedy Matching Comparison on Manually Annotated Dataset for Location

metric the performed better in terms of recall is Jaro-Winkler.

Top-down Rules for Location

In tables 10.10 10.13 we presented the results of matching performances of the top down matching rules defined. The performances on the weighted average performance on the manually annotated dataset are quite good for locations. The large number of non matching cases were identifier. However, the poor recall of positive labeled sample is a sign that probably the rules defined are too restrictive to support matching decisions. Equal similarity metric produced poor results, reflecting the syntactical heterogeneity affecting representation of attributes values. The overall $\rho - accuracy$ is aligned with the weighted average score. This is due probably to the fact that the number of negative sample is way to large than the one positive matching samples. However, all the similarity metrics produced similar results with greedy approach to comparison. The results of the evaluation of the same dataset with Knowledge-Driven and Character-based Relative Completeness comparison methods produced the pretty much the same results.

It is important to notice how all the similarity metrics anyway matched with a high level of precision.

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.056	0.056	0.056	0.056	0.056	0.111	0.056	0.056	0.056	0.056	0.056
	F-Measure	0.105	0.105	0.105	0.105	0.105	0.2	0.105	0.105	0.105	0.105	0.105
NonMatch	Precision	0.929	0.944	0.944	0.944	0.944	0.946	0.944	0.944	0.946	0.944	0.944
	Recall	0.033	1.0	1.0	1.0	1.0	0.997	1.0	1.0	0.997	1.0	1.0
	F-Measure	0.065	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971
DontKnow	Precision	0.015	NaN	NaN	NaN	NaN	0.0	NaN	NaN	0.0	NaN	NaN
	Recall	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	F-Measure	0.03	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Weighted Avg	Precision	0.918	NaN	NaN	NaN	NaN	0.935	NaN	NaN	0.935	NaN	NaN
	Recall	0.048	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.942	0.944	0.944
	F-Measure	0.066	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	ρ-Accuracy	0.092	0.944	0.944	0.944	0.944	0.946	0.944	0.944	0.944	0.944	0.944

Table 10.11: Knowledge-driven Comparison Method on Manually Annotated Dataset for Location

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.056	0.056	0.056	0.056	0.056	0.111	0.056	0.056	0.056	0.056	0.056
	F-Measure	0.105	0.105	0.105	0.105	0.105	0.2	0.105	0.105	0.105	0.105	0.105
NonMatch	Precision	0.929	0.944	0.944	0.944	0.944	0.946	0.944	0.944	0.946	0.944	0.944
	Recall	0.033	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	F-Measure	0.065	0.971	0.971	0.971	0.971	0.972	0.971	0.971	0.972	0.971	0.971
DontKnow	Precision	0.015	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN
	Recall	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	F-Measure	0.03	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Weighted Avg	Precision	0.918	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.935	NaN	NaN
	Recall	0.048	0.944	0.944	0.944	0.944	0.947	0.944	0.944	0.944	0.944	0.944
	F-Measure	0.066	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	ρ-Accuracy	0.092	0.944	0.944	0.944	0.944	0.947	0.944	0.944	0.946	0.944	0.944

Table 10.12: Character-Based RC Estimator Comparison on Manually Annotated Dataset for Location

Matching performances on NYT dataset for location are quite poor using this set of top down matching rules. Positive matching decisions were taken with high precision, but very seldom unfortunately. The best similarity metric, although in a context of very low recall, is Monge-Elkan. A possible interpretation of this very low recall has to be referred to the syntactical heterogeneity affecting the representation attributes such as latitude and longitude. However, another interpretation is that the negative matching rules were too relaxed, causing a large number of false negative matching. In fact,, also ρ accuracy is lower than the f-measure, highlighting bad matching performances.

Top-down Rules for Organization

In tables 10.14, 10.16, 10.17 we presented the results of matching performances of the top down matching rules defined. The performances on the weighted average performance on the manually annotated dataset are quite good. This is due mostly to the large number of negative matching samples, which are generally classified correctly. Equal similarity metric produced poor results, as a sign of the syntactic heterogeneity affecting representation of attributes values. The positive matching performances are quite poor. The number of positive matching samples is quite low, and apparently hard

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Greedy	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.018	0.03	0.03	0.03	0.03	0.222	0.03	0.03	0.04	0.03	0.049
	F-Measure	0.035	0.057	0.057	0.057	0.057	0.364	0.057	0.058	0.077	0.057	0.094
	ρ-Accuracy	0.019	0.032	0.032	0.032	0.032	0.232	0.032	0.032	0.043	0.032	0.051
Knowledge	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.018	0.018	0.018	0.018	0.018	0.291	0.018	0.018	0.024	0.018	0.029
	F-Measure	0.035	0.035	0.035	0.035	0.035	0.451	0.035	0.036	0.047	0.035	0.057
	ρ-Accuracy	0.029	0.019	0.019	0.019	0.019	0.307	0.019	0.019	0.026	0.019	0.031
Char RC	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.03	0.021	0.03	0.03	0.012	0.117	0.03	0.023	0.02	0.017	0.03
	F-Measure	0.057	0.041	0.057	0.057	0.023	0.21	0.057	0.045	0.04	0.033	0.057
	ρ-Accuracy	0.048	0.022	0.032	0.032	0.013	0.123	0.032	0.024	0.022	0.018	0.032

Table 10.13: Matching Experiment on NYT Dataset for Location

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.1	0.1	0.1	0.1	0.071	0.121	0.182	0.1	0.133	0.091	0.091
	Recall	0.029	0.029	0.029	0.029	0.029	0.118	0.059	0.029	0.059	0.029	0.029
	F-Measure	0.045	0.045	0.045	0.045	0.042	0.119	0.089	0.045	0.082	0.044	0.044
NonMatch	Precision	0.822	0.978	0.978	0.978	0.978	0.979	0.982	0.978	0.985	0.978	0.978
	Recall	0.121	0.975	0.975	0.975	0.973	0.916	0.913	0.975	0.905	0.975	0.975
	F-Measure	0.211	0.976	0.976	0.976	0.976	0.947	0.946	0.976	0.943	0.976	0.976
DontKnow	Precision	0.011	0.135	0.135	0.135	0.135	0.062	0.071	0.135	0.065	0.135	0.135
	Recall	0.52	0.28	0.28	0.28	0.28	0.28	0.4	0.28	0.4	0.28	0.28
	F-Measure	0.021	0.182	0.182	0.182	0.182	0.102	0.12	0.182	0.112	0.182	0.182
Weighted Avg	Precision	0.79	0.941	0.941	0.941	0.941	0.942	0.946	0.941	0.947	0.941	0.941
	Recall	0.126	0.94	0.94	0.94	0.938	0.886	0.884	0.94	0.875	0.939	0.939
	F-Measure	0.203	0.94	0.94	0.94	0.939	0.912	0.911	0.94	0.907	0.939	0.939
	ρ-Accuracy	0.216	0.95	0.95	0.95	0.945	0.9	0.922	0.95	0.913	0.949	0.949

Table 10.14: Greedy Matching Comparison on Manually Annotated Dataset for Organization

to guess with the defined rules. The very low precision in the DontKnow classification seems to suggest that many positive matching examples were classified as unknown, which might imply that match similarity threshold defined as to 0.9 is too restrictive for most of similarity metrics. However, the similarity metrics performing worst are Monge-Elkan, Overlap and Jaro-Winkler, even thou the overall performances are still pretty decent. The results of the evaluation of the same dataset with Knowledge-Driven comparison methods produced the same results. Different results were produce using the Character-Based Relative Completeness (RC) comparison method presented in table 10.15. This comparison method produces slightly better results for all the similarity metrics, allowing a better classification of negative matching samples. Also TagLink and Needlman Wunsch emerged as effective similarity metrics. In fact, relative completeness reduces the problems related to a high number of incomplete variations of attributes that could affect negatively greedy matching comparison.

More problematic appears the evaluation of the experiment on the New York Times dataset (table 10.16) where only two similarity metric produces scores sufficient to produce positive matching decision on the analyzed descriptions. In fact, only Monge-Elkan and Smith-Waterman could produce some matching measure above the considered thresholds considered all the three comparison methods. Considers that the NYT

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.1	0.1	0.1	0.1	0.071	0.167	0.182	0.1	0.133	0.091	0.091
	Recall	0.029	0.029	0.029	0.029	0.029	0.118	0.059	0.029	0.059	0.029	0.029
	F-Measure	0.045	0.045	0.045	0.045	0.042	0.138	0.089	0.045	0.082	0.044	0.044
NonMatch	Precision	0.822	0.978	0.977	0.978	0.978	0.979	0.982	0.978	0.984	0.977	0.977
	Recall	0.121	0.975	0.981	0.975	0.979	0.929	0.913	0.975	0.911	0.981	0.981
	F-Measure	0.211	0.976	0.979	0.976	0.978	0.953	0.946	0.976	0.946	0.979	0.979
DontKnow	Precision	0.011	0.135	0.14	0.135	0.14	0.058	0.071	0.135	0.062	0.14	0.14
	Recall	0.52	0.28	0.24	0.28	0.24	0.24	0.4	0.28	0.36	0.24	0.24
	F-Measure	0.021	0.182	0.176	0.182	0.176	0.094	0.12	0.182	0.107	0.176	0.176
Weighted Avg	Precision	0.79	0.941	0.941	0.941	0.941	0.943	0.946	0.941	0.947	0.94	0.94
	Recall	0.126	0.94	0.945	0.94	0.943	0.897	0.884	0.94	0.88	0.944	0.944
	F-Measure	0.203	0.94	0.942	0.94	0.941	0.918	0.911	0.94	0.91	0.942	0.942
	ρ-Accuracy	0.216	0.95	0.953	0.95	0.948	0.917	0.922	0.95	0.917	0.951	0.951

Table 10.15: Character-Based RC Estimator Comparison on Manually Annotated Dataset for Organization

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Greedy	Precision	NaN	NaN	NaN	NaN	NaN	1.0	NaN	NaN	1.0	NaN	NaN
	Recall	0.0	0.0	0.0	0.0	0.0	0.239	0.0	0.0	0.016	0.0	0.0
	F-Measure	NaN	NaN	NaN	NaN	NaN	0.386	NaN	NaN	0.031	NaN	NaN
	ρ-Accuracy	0.0	0.006	0.006	0.006	0.006	0.301	0.007	0.006	0.03	0.006	0.006
Knowledge	Precision	NaN	NaN	NaN	NaN	NaN	1.0	NaN	NaN	1.0	NaN	NaN
	Recall	0.0	0.0	0.0	0.0	0.0	0.355	0.0	0.0	0.022	0.0	0.0
	F-Measure	NaN	NaN	NaN	NaN	NaN	**0.524**	NaN	NaN	0.043	NaN	NaN
RC Char	ρ-Accuracy	0.0	0.006	0.006	0.006	0.006	0.429	0.007	0.006	0.038	0.006	0.006
	Precision	NaN	NaN	NaN	NaN	NaN	1.0	NaN	NaN	1.0	NaN	NaN
	Recall	0.0	0.0	0.0	0.0	0.0	0.134	0.0	0.0	0.01	0.0	0.0
	F-Measure	NaN	NaN	NaN	NaN	NaN	0.237	NaN	NaN	0.019	NaN	NaN
	ρ-Accuracy	0.0	0.006	0.006	0.006	0.006	0.176	0.007	0.006	0.021	0.006	0.006

Table 10.16: Matching Experiment on NYT Dataset for Organization

dataset contains only positive matching samples, and thus we cannot evaluate negative match classification and thus we present all the comparison methods in the same table. Knowledge-Driven method applies greedy approach on functional diachronic attributes, and average on others. However, it may still be interesting to discern the similarity method that produced less false negative match decision, choosing rather to classify as DontKnow. Also considering this score, Monge-Elkan is the one that produced a higher number of true positives and reduced the number of false negative match.

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Greedy	Precision	NaN	NaN	NaN	NaN	NaN	1.0	NaN	NaN	NaN	NaN	1.0
	Recall	0.0	0.0	0.0	0.0	0.0	0.034	0.0	0.0	0.0	0.0	0.854
	F-Measure	NaN	NaN	NaN	NaN	NaN	0.065	NaN	NaN	NaN	NaN	**0.921**
	ρ-Accuracy	0.0	0.0	0.0	0.0	0.0	0.053	0.0	0.0	0.0	0.0	0.863
Knowledge	Precision	NaN	NaN	NaN	NaN	NaN	1.0	NaN	NaN	NaN	NaN	1.0
	Recall	0.0	0.0	0.0	0.0	0.0	0.034	0.0	0.0	0.0	0.0	0.854
	F-Measure	NaN	NaN	NaN	NaN	NaN	0.065	NaN	NaN	NaN	NaN	0.921
	ρ-Accuracy	0.0	0.0	0.0	0.0	0.0	0.053	0.0	0.0	0.0	0.0	0.863
Char RC	Precision	NaN	NaN	NaN	NaN	NaN	1.0	NaN	NaN	NaN	NaN	1.0
	Recall	0.0	0.0	0.0	0.0	0.0	0.034	0.0	0.0	0.0	0.0	0.854
	F-Measure	NaN	NaN	NaN	NaN	NaN	0.065	NaN	NaN	NaN	NaN	0.921
	ρ-Accuracy	0.0	0.0	0.0	0.0	0.0	0.053	0.0	0.0	0.0	0.0	0.863

Table 10.17: Matching Comparison on OAEI 2010 Dataset for Organization

Also the evaluation of the OAEI 2010 dataset about restaurants (table 10.17) produced disappointing results. However, Taglink similarity metric produced outstanding results, showing to cope well with the syntactic variations of descriptions. Also considering the score defined in function (10.1), the best comparison method and similarity metric are Greedy Comparison Method and Taglink.

10.2.2 Bottom-up Only Rules Experiments

In this section we evaluate the matching method we proposed in this thesis relying only on bottom-learned rules. Hence, this section will present a list of experiments related to the evaluation of factors that can affect learning process, and the combination of rules extracted. Essentially, the learning process is the following:

1. we gathered samples of data from heterogeneous sources;
2. we defined contextual mappings to ease the problem of semantic harmonization;
3. we defined a blocking scheme to select description pairs to be labeled;
4. we labeled descriptions pairs as match, non-match and don't know.
5. we use the labeled set to extract matching rules using decision tree classifiers;
6. we implemented a set of merging and normalization processes for extracted rules;

The learning process is described in detail in section 8.2.

The experiments will then have to consider several dimensions. Clearly, matching comparison methods affect quality of learned rules, as a greedy approach may produce higher score in the matching of syntactically heterogeneous attributes also when they are not matching, and thus affect evaluation of the relevance of that attribute along the learning process. Clearly, also string similarity metrics affect the learning process, as the more a metric is capable of representing clear distinction between matching and non-matching attributes, the more precise the thresholds embedded in the rules would be. Then, other aspects come from the training set partition and filtering. In fact, we want to experimentally evaluate the impact of the 'don't know' labeled samples in the learning process, as two and three class classification process may produce different results. Furthermore, we have to evaluate the normalization processes. In particular, we focus on the inconsistency normalization, testing how removing inconsistent atoms and normalizing inconsistent atoms affects the decision process. Finally, we also evaluate the impact of the thresholds normalization, choosing the most conservative and relaxed thresholds for both positive and negative matching rules. Therefore, in the following, we are going to propose a quite extensive set of experiment results. However, for the mere sake of keeping the presentation compact, we avoid presenting table results which do not present variants in the final results. These less informative tables will be anyway made available on the author website, together with the raw experiment results and datasets.

Bottom-up Rules for Person

Evaluating learned rules on the manually annotated dataset considering binary classification method and greedy comparison approach produced the best accuracy considering the inconsistency removal normalization process, independently from the conservative or relaxed threshold selection, relying on Monge-Elkan similarity metric (table 10.18). Notice that also rules learned relying on similarity metrics Needlman-Wunsch allowed to match with high accuracy. Notice how the ρ-accuracy scores are higher than the weighted f-measure, meaning that in general, rather than take inaccurate matching decisions, the rules supported more conservative don't know classification. However, Monge-Elkan was not the best in terms of precision when it came to classify positive matching samples. In regards to this, Taglink is the similarity metric that supported higher precision in the considered settings. The substance of the evaluation does not change if we consider multiclass classification (table 10.19), as the best matching accuracy is achieved relying on the same configuration.

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.927	0.933	1.0	1.0	0.925	0.944	0.917	0.898	1.0	0.944	1.0
	Recall	0.389	0.321	0.115	0.176	0.374	0.13	0.084	0.336	0.153	0.389	0.153
	F-Measure	0.548	0.477	0.205	0.299	0.533	0.228	0.154	0.489	0.265	0.551	0.265
NonMatch	Precision	0.165	0.986	0.995	0.993	0.975	0.941	0.956	0.983	0.951	0.973	0.999
	Recall	0.008	0.498	0.684	0.66	0.786	0.923	0.846	0.418	0.824	0.856	0.475
	F-Measure	0.016	0.661	0.811	0.793	0.87	0.932	0.898	0.586	0.883	0.911	0.644
DontKnow	Precision	0.032	0.075	0.113	0.109	0.139	0.204	0.137	0.066	0.165	0.167	0.077
	Recall	0.682	0.909	0.989	1.0	0.818	0.545	0.648	0.909	0.83	0.727	1.0
	F-Measure	0.061	0.138	0.202	0.197	0.238	0.297	0.226	0.123	0.275	0.272	0.142
Weighted Avg	Precision	0.208	0.943	0.957	0.956	0.935	0.909	0.918	0.938	0.92	0.936	0.959
	Recall	0.062	0.504	0.66	0.643	0.761	0.856	0.788	0.434	0.781	0.821	0.477
	F-Measure	0.052	0.627	0.745	0.735	0.821	0.859	0.821	0.56	0.817	0.86	0.598
	ρ-Accuracy	0.111	0.667	0.794	0.781	0.852	**0.897**	0.865	0.6	0.859	0.89	0.646

Table 10.18: TCBI SCALL IPP NPIR TNCC Greedy for Person

If we consider Knowledge-driven matching comparison methods, the configuration that performed better is the one that applied inconsistencies removal normalization, independently from the chosen thresholds and relying on the Needleman-Wunsch string similarity metric (table 10.20). If we consider multi-class classification, the performances decrease in terms of accuracy, but the precision of positive match seem to be improved, despite the general decrease in recall. In this case, the similarity metric performing better is Monge-Elkan (table 10.21), that anyway was among the best also considering binary classification.

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.927	0.933	1.0	0.93	0.907	0.9	1.0	0.896	1.0	0.949	0.91
	Recall	0.389	0.321	0.183	0.305	0.298	0.206	0.046	0.328	0.206	0.282	0.466
	F-Measure	0.548	0.477	0.31	0.46	0.448	0.335	0.088	0.48	0.342	0.435	0.616
NonMatch	Precision	0.165	0.985	0.981	0.992	0.989	0.949	0.967	0.993	0.95	0.991	0.986
	Recall	0.008	0.556	0.146	0.673	0.153	0.898	0.763	0.146	0.721	0.175	0.435
	F-Measure	0.016	0.711	0.254	0.802	0.264	0.923	0.853	0.255	0.82	0.298	0.604
DontKnow	Precision	0.032	0.083	0.049	0.111	0.049	0.181	0.111	0.049	0.105	0.051	0.07
	Recall	0.682	0.909	0.977	0.955	0.955	0.591	0.75	0.955	0.75	0.977	0.932
	F-Measure	0.061	0.152	0.094	0.198	0.093	0.277	0.193	0.093	0.184	0.098	0.131
Weighted Avg	Precision	0.208	0.943	0.942	0.95	0.943	0.913	0.932	0.945	0.917	0.947	0.942
	Recall	0.062	0.556	0.184	0.662	0.197	0.84	0.716	0.193	0.689	0.217	0.459
	F-Measure	0.052	0.672	0.251	0.754	0.269	0.857	0.775	0.262	0.762	0.298	0.584
	ρ-Accuracy	0.111	0.711	0.311	0.793	0.326	**0.891**	0.823	0.321	0.8	0.354	0.623

Table 10.19: TCMU SCALL NPIR TNCC Greedy Manually Annotated Dataset for Person

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.927	0.933	1.0	1.0	0.925	1.0	0.941	0.912	1.0	0.944	0.941
	Recall	0.389	0.321	0.115	0.176	0.374	0.008	0.122	0.237	0.16	0.389	0.366
	F-Measure	0.548	0.477	0.205	0.299	0.533	0.015	0.216	0.376	0.276	0.551	0.527
NonMatch	Precision	0.165	0.986	0.995	0.994	0.975	0.945	0.957	0.982	0.952	0.959	0.995
	Recall	0.008	0.498	0.684	0.657	0.786	0.913	0.831	0.426	0.824	0.913	0.511
	F-Measure	0.016	0.661	0.811	0.791	0.87	0.929	0.889	0.595	0.883	0.936	0.675
DontKnow	Precision	0.032	0.075	0.113	0.109	0.139	0.19	0.134	0.067	0.165	0.155	0.08
	Recall	0.682	0.909	0.989	1.0	0.818	0.602	0.67	0.92	0.83	0.443	0.955
	F-Measure	0.061	0.138	0.202	0.196	0.237	0.289	0.223	0.125	0.275	0.229	0.147
Weighted Avg	Precision	0.208	0.943	0.957	0.956	0.936	0.916	0.92	0.938	0.921	0.924	0.952
	Recall	0.062	0.504	0.66	0.641	0.761	0.842	0.778	0.436	0.781	0.859	0.521
	F-Measure	0.052	0.627	0.745	0.734	0.821	0.843	0.817	0.56	0.818	0.881	0.643
	ρ-Accuracy	0.111	0.667	0.794	0.78	0.852	0.891	0.86	0.603	0.86	**0.907**	0.683

Table 10.20: TCBI SCALL IPP NPIR TNCC Manually Annotated Dataset for Person

If we consider character-based relative completeness to weight greedy compared rules obtained relying on binary classification, the configuration the performed better in terms of ρ-accuracy is the one that applied inconsistency removal for inconsistencies normalization, defined relaxed threshold for positive matching rules and conservative thresholds for negative matching rules, relying on Overlap similarity metric (table 10.22). The overall performance on positive matching seems to be improved with respect to the other configurations analyzed so far, but at the same time, non matching performance decreased. If we consider multiclass classification, the configuration that performed better in terms of ρ-accuracy is the same as for binary classification, but considering Jaccard string similarity metric (table 10.23).

Performing experiments with the New York Times dataset for person considering binary classification method, the configuration that performed better is the one that applied inconsistencies removal normalization (NPIR), and applied Relaxed thresholds for positive matching rules and conservative thresholds for negative matching rules, using Monge-Elkan string similarity metric and relying on a comparison method that weighted greedy attribute comparison score with Character-based Relative Complete-

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.927	0.933	1.0	0.93	0.907	1.0	NaN	0.896	1.0	0.944	1.0
	Recall	0.389	0.321	0.183	0.305	0.298	0.008	0.0	0.328	0.206	0.26	0.183
	F-Measure	0.548	0.477	0.31	0.46	0.448	0.015	NaN	0.48	0.342	0.407	0.31
NonMatch	Precision	0.165	0.985	0.988	0.992	0.989	0.964	0.957	0.993	0.949	0.989	0.999
	Recall	0.008	0.556	0.139	0.673	0.153	0.866	0.783	0.148	0.725	0.192	0.445
	F-Measure	0.016	0.711	0.244	0.802	0.264	0.912	0.862	0.257	0.822	0.322	0.615
DontKnow	Precision	0.032	0.083	0.05	0.111	0.049	0.144	0.115	0.049	0.116	0.052	0.073
	Recall	0.682	0.909	1.0	0.955	0.955	0.659	0.716	0.955	0.818	0.966	1.0
	F-Measure	0.061	0.152	0.096	0.198	0.093	0.236	0.198	0.093	0.203	0.098	0.136
Weighted Avg	Precision	0.208	0.943	0.948	0.95	0.943	0.931	NaN	0.946	0.916	0.945	0.959
	Recall	0.062	0.556	0.179	0.662	0.197	0.801	0.73	0.194	0.695	0.23	0.452
	F-Measure	0.052	0.672	0.242	0.754	0.269	0.825	NaN	0.264	0.764	0.318	0.575
	ρ-Accuracy	0.111	0.711	0.304	0.793	0.326	**0.875**	0.83	0.323	0.803	0.373	0.623

Table 10.21: TCMU SCALL IPP NPIR TNCC Manually Annotated Dataset for Person

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.941	0.926	0.956	1.0	0.931	0.848	0.983	0.918	0.975	0.718	1.0
	Recall	0.366	0.382	0.328	0.176	0.206	0.511	0.45	0.427	0.298	0.603	0.275
	F-Measure	0.527	0.541	0.489	0.299	0.338	0.638	0.618	0.583	0.456	0.656	0.431
NonMatch	Precision	0.174	0.977	0.983	0.992	0.992	0.992	0.987	0.985	0.964	0.972	0.991
	Recall	0.008	0.586	0.784	0.681	0.218	0.704	0.779	0.671	0.679	0.769	0.767
	F-Measure	0.016	0.732	0.872	0.808	0.357	0.823	0.871	0.798	0.797	0.858	0.865
DontKnow	Precision	0.032	0.084	0.14	0.115	0.052	0.118	0.14	0.104	0.088	0.115	0.133
	Recall	0.682	0.852	0.864	1.0	0.955	0.898	0.864	0.875	0.716	0.636	0.898
	F-Measure	0.06	0.153	0.242	0.206	0.099	0.209	0.242	0.186	0.157	0.194	0.232
Weighted Avg	Precision	0.218	0.935	0.945	0.955	0.948	0.945	0.95	0.943	0.927	0.919	0.955
	Recall	0.06	0.584	0.758	0.662	0.249	0.7	0.762	0.664	0.656	0.752	0.741
	F-Measure	0.051	0.695	0.82	0.749	0.345	0.785	0.827	0.758	0.747	0.817	0.81
	ρ-Accuracy	0.109	0.729	0.855	0.794	0.396	0.812	**0.86**	0.791	0.781	0.828	0.848

Table 10.22: TCBI SCALL NPIR TNRC Char-RC Manually Annotated Dataset for Person

ness (Char-RC) (table 10.24). Notice that the ρ-accuracy is lower than the f-measure, meaning that some of samples was classified as negative matches. Considering multiclass classification including "don't know" labeled samples, the best result was obtained with the same configuration as for binary classification, but using the Smith-Waterman string similarity metric. The overall accuracy is little lower, but the difference between the f-measure and the accuracy is lower than in the case of binary classification (table 10.25). This means that multiclass classification supported a more conservative classification of negative matching samples. The fact that Relative Completeness weighted similarity metrics supported learning of more accurate matching rules indicates that syntactical variations on attribute values allow to learn more relaxed thresholds increasing the recall of positive matches.

Performing experiments on the OAEI 2010 dataset for person, the configuration that allowed extracting more effective rules better in terms of ρ-accuracy considering binary classification method is the one that applied inconsistency removal normalization, applied relaxed threshold for positive matching rules and conservative threshold for negative matching rules, relying on Needlman-Wunsch similarity metric (table

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.925	0.559	0.697	0.96	0.978	0.917	0.952	0.957	0.938	0.814	1.0
	Recall	0.282	0.145	0.176	0.183	0.344	0.42	0.458	0.336	0.458	0.366	0.237
	F-Measure	0.433	0.23	0.28	0.308	0.508	0.576	0.619	0.497	0.615	0.505	0.383
NonMatch	Precision	0.155	0.991	0.958	0.981	0.988	0.962	0.992	0.988	0.995	0.976	0.999
	Recall	0.008	0.56	0.582	0.814	0.441	0.721	0.706	0.349	0.615	0.664	0.629
	F-Measure	0.016	0.715	0.724	0.889	0.61	0.824	0.825	0.515	0.76	0.79	0.772
DontKnow	Precision	0.032	0.08	0.066	0.151	0.074	0.106	0.122	0.062	0.103	0.104	0.102
	Recall	0.682	0.886	0.67	0.864	0.989	0.739	0.943	0.955	0.989	0.875	1.0
	F-Measure	0.06	0.147	0.12	0.257	0.138	0.185	0.216	0.117	0.186	0.186	0.186
Weighted Avg	Precision	0.199	0.924	0.903	0.944	0.948	0.922	0.952	0.945	0.952	0.928	0.96
	Recall	0.055	0.547	0.559	0.775	0.459	0.703	0.7	0.374	0.621	0.654	0.62
	F-Measure	0.044	0.66	0.669	0.825	0.583	0.781	0.785	0.497	0.726	0.746	0.722
	ρ-Accuracy	0.099	0.696	0.7	**0.867**	0.627	0.81	0.82	0.542	0.762	0.777	0.765

Table 10.23: TCMU SCALL NPIR TNCR Char-RC Manually Annotated Dataset for Person

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Greedy	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.355	0.757	0.002	0.614	0.204	0.424	0.032	0.649	0.555	0.414	0.597
	F-Measure	0.524	0.861	0.004	0.761	0.339	0.595	0.063	0.787	0.714	0.585	0.748
	ρ-Accuracy	0.523	**0.854**	0.004	0.751	0.239	0.595	0.062	0.752	0.712	0.521	0.745
Knowledge	Precision	1.0	1.0	1.0	1.0	1.0	NaN	1.0	1.0	1.0	1.0	1.0
	Recall	0.355	0.757	0.003	0.688	0.791	0.0	0.031	0.327	0.609	0.322	0.195
	F-Measure	0.524	0.861	0.006	0.815	0.884	NaN	0.061	0.493	0.757	0.488	0.327
	ρ-Accuracy	0.523	0.854	0.006	0.814	**0.882**	0.0	0.06	0.492	0.756	0.406	0.324
Char-RC	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.73	0.642	0.638	0.396	0.395	0.874	0.767	0.775	0.755	0.769	0.68
	F-Measure	0.844	0.782	0.779	0.568	0.566	0.933	0.868	0.873	0.86	0.87	0.809
	ρ-Accuracy	0.809	0.765	0.734	0.528	0.559	**0.906**	0.83	0.828	0.822	0.796	0.792

Table 10.24: TCBI SCALL TNCC NYT Dataset for Person

tab:bottomup-oaei-binary-person). Considering multiclass classification method, the best configuration is the one that applied greedy comparison method, applying inconsistency removal and relaxed threshold for matching, and relying on the qgram similarity metric. 10.27.

Bottom-up Rules for Location

In this section we present the experiments that obtained the best results on locations evaluation datasets. In particular, for each comparison method, and classification method, we select the best inconsistency normalization and threshold normalization combination based on the ρ-accuracy described in equation (10.1) at the beginning of the chapter. The detailed results of the other experiments will be available online, together with the raw experiment results and the datasets.

Considering a greedy comparison approach (Greedy), and rules extracted relying on binary classification (TCBI) considering all the sources at the same time (SCALL), the configuration that produced the best results on the manually annotated dataset in terms of ρ-accuracy is the one that applied inconsistency removal normalization (NPIR), and Conservative Match Relaxed Non Match threshold normalization (TCCR),

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Greedy	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.355	0.754	0.464	0.717	0.011	0.556	0.001	0.74	0.47	0.468	0.01
	F-Measure	0.524	0.86	0.634	0.835	0.023	0.715	0.002	0.85	0.64	0.638	0.02
	ρ-Accuracy	0.523	**0.859**	0.633	0.83	0.022	0.713	0.002	0.791	0.639	0.496	0.02
Knowledge	Precision	1.0	1.0	1.0	1.0	1.0	NaN	1.0	1.0	1.0	1.0	1.0
	Recall	0.355	0.754	0.474	0.759	0.009	0.0	0.028	0.745	0.317	0.039	0.004
	F-Measure	0.524	0.86	0.643	0.863	0.018	NaN	0.054	0.854	0.481	0.074	0.008
	ρ-Accuracy	0.523	0.836	0.643	*0.861*	0.018	0.0	0.054	0.797	0.481	0.075	0.007
Char-RC	Precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.6	0.119	0.305	0.401	0.702	0.779	0.761	0.744	0.855	0.659	0.687
	F-Measure	0.75	0.213	0.468	0.572	0.825	0.876	0.864	0.853	0.922	0.794	0.814
	ρ-Accuracy	0.664	0.211	0.465	0.528	0.783	0.817	0.811	0.835	**0.898**	0.783	0.804

Table 10.25: TCMU SCALL IPP NPIC TNRC NYT Dataset for Person

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Greedy	Precision	1.0	1.0	NaN	NaN	1.0	NaN	NaN	1.0	NaN	1.0	NaN
	Recall	0.341	0.486	0.0	0.0	0.43	0.0	0.0	0.61	0.0	0.508	0.0
	F-Measure	0.509	0.654	NaN	NaN	0.601	NaN	NaN	0.758	NaN	0.674	NaN
	ρ-Accuracy	0.38	0.636	0.0	0.0	0.499	0.0	0.0	**0.743**	0.0	0.576	0.0
Knowledge	Precision	1.0	1.0	NaN	NaN	1.0	NaN	NaN	1.0	NaN	1.0	NaN
	Recall	0.321	0.478	0.0	0.0	0.62	0.0	0.0	0.001	0.0	0.01	0.0
	F-Measure	0.486	0.647	NaN	NaN	0.765	NaN	NaN	0.002	NaN	0.02	NaN
	ρ-Accuracy	0.486	0.644	0.0	0.0	**0.759**	0.0	0.0	0.002	0.0	0.019	0.0
Char-RC	Precision	1.0	1.0	1.0	NaN	1.0	1.0	NaN	1.0	NaN	1.0	NaN
	Recall	0.338	0.529	0.326	0.0	0.399	0.387	0.0	0.538	0.0	0.758	0.0
	F-Measure	0.505	0.692	0.491	NaN	0.57	0.558	NaN	0.699	NaN	0.862	NaN
	ρ-Accuracy	0.461	0.658	0.364	0.0	0.568	0.521	0.0	0.668	0.0	**0.795**	0.0

Table 10.26: TCBI SCALL NPIR TNRC OAEI 2010 for Person

using Jaro-Winkler similarity metric. The detailed results are presented in table 10.28. It is interesting to see how also Levenshtein and Taglink performed well. However, using Taglink some positive match was discovered, whereas Levenshtein was just more effective in increasing the recall of negative matching decisions.

If we consider the same settings, but learn rules considering also the third "don't know" class, the configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistency removal normalization (NPIR), applying relaxed thresholds both on atoms of positive matching and negative matching rules (TCRR), using Levenshtein similarity metric. The details of the experiments with this configuration are presented in table 10.29. In this case, also QGram performed well, whereas Taglink lost precision in the matching classification. The increasing complexity related to the don't know cases did not support a general improvement in terms of overall ρ-accuracy, but reduced in the precision of the positive match decision for some metrics.

Considering a Knowledge-driven comparison approach (Knowledge), and rules extracted relying on binary classification (TCBI) considering all the sources at the same time (SCALL), the configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistency removal normalization (NPIR), independently from the threshold normalization function, using QGram similarity metric. The de-

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	1.0	1.0	NaN	1.0	1.0	NaN	NaN	1.0	NaN	1.0	NaN
	Recall	0.341	0.458	0.0	0.341	0.43	0.0	0.0	0.61	0.0	0.436	0.0
	F-Measure	0.509	0.628	NaN	0.509	0.601	NaN	NaN	0.758	NaN	0.607	NaN
	ρ-Accuracy	0.38	0.608	0.0	0.423	0.589	0.0	0.0	**0.746**	0.0	0.595	0.0
Match	Precision	1.0	1.0	NaN	1.0	1.0	NaN	NaN	1.0	NaN	1.0	NaN
	Recall	0.341	0.458	0.0	0.341	0.342	0.0	0.0	0.379	0.0	0.436	0.0
	F-Measure	0.509	0.628	NaN	0.509	0.51	NaN	NaN	0.55	NaN	0.607	NaN
	ρ-Accuracy	0.38	0.512	0.0	0.423	0.499	0.0	0.0	0.539	0.0	0.533	0.0
Match	Precision	1.0	NaN	NaN	NaN	1.0	1.0	NaN	1.0	NaN	1.0	1.0
	Recall	0.321	0.0	0.0	0.0	0.476	0.392	0.0	0.34	0.0	0.48	0.38
	F-Measure	0.486	NaN	NaN	NaN	0.645	0.563	NaN	0.507	NaN	0.649	0.551
	ρ-Accuracy	0.438	0.0	0.0	0.0	0.631	0.543	0.0	0.506	0.0	0.649	0.538

Table 10.27: TCMU SCALL IPP NPIR TNRC OAEI 2010 for Person

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	NaN	NaN	0.2	NaN	1.0	0.5	NaN	1.0	0.667	0.545	1.0
	Recall	0.0	0.0	0.722	0.0	0.167	0.056	0.0	0.167	0.111	0.333	0.056
	F-Measure	NaN	NaN	0.313	NaN	0.286	0.1	NaN	0.286	0.19	0.414	0.105
NonMatch	Precision	NaN	0.984	0.977	NaN	0.977	0.984	0.944	0.987	0.979	0.96	0.992
	Recall	0.0	0.964	0.874	0.0	0.985	0.774	0.997	0.949	0.956	0.987	0.949
	F-Measure	NaN	0.974	0.923	NaN	0.981	0.866	0.97	0.967	0.967	0.973	0.97
DontKnow	Precision	0.015	0.094	NaN	0.015	0.056	0.038	0.0	0.028	0.033	0.0	0.075
	Recall	1.0	0.5	0.0	1.0	0.167	0.667	0.0	0.167	0.167	0.0	0.5
	F-Measure	0.029	0.158	NaN	0.029	0.083	0.072	NaN	0.048	0.056	NaN	0.13
Weighted Avg	Precision	NaN	NaN	NaN	NaN	0.965	0.949	NaN	0.973	0.952	0.928	0.979
	Recall	0.015	0.915	0.855	0.015	0.937	0.741	0.939	0.903	0.908	0.944	0.903
	F-Measure	NaN	NaN	NaN	NaN	0.937	0.821	NaN	0.924	0.92	NaN	0.92
	ρ-Accuracy	0.03	0.948	0.759	0.03	**0.957**	0.843	0.941	0.943	0.94	0.935	0.945

Table 10.28: TCBI SCALL NPIR TNCR Greedy on Manually Annotated dataset for Location

tailed results are presented in table 10.30. It is interesting to see how also Taglink and Euclidean similarity metrics performed well. However, also in this case using Taglink some positive match was discovered, whereas Euclidean was just more effective in increasing the recall of negative matching decisions. It is important to notice that the QGram did not support very precise positive matching decisions, although the recall was the best one. Also Monge-Elkan similarity metric supported a high recall in the number of matched pairs, however, the precision was not as good as Taglink and QGram.

If we consider the same settings, but learn rules considering also the third "don't know" class (TCMU), the configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistency removal normalization (NPIR), applying relaxed thresholds both on atoms of positive matching and negative matching rules (TCRR), using Taglink similarity metric. Also in this case, considering multiclass classification, rules learned and applied using Taglink reduced precision performances, but effectively classified a large number of positive matching samples, and nearly all negative matching samples. Also rules learned and applied using Levenshtein similarity metric produced very good results, with a better precision in supporting matching decision, but with a lower recall.

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	NaN	0.857	0.2	NaN	1.0	0.5	NaN	1.0	0.667	0.714	0.8
	Recall	0.0	0.667	0.722	0.0	0.056	0.056	0.0	0.056	0.111	0.278	0.667
	F-Measure	NaN	0.75	0.313	NaN	0.105	0.1	NaN	0.105	0.19	0.4	0.727
NonMatch	Precision	NaN	0.977	0.977	0.944	1.0	0.977	NaN	0.975	0.991	0.962	0.996
	Recall	0.0	0.977	0.874	0.997	0.003	0.964	0.0	0.985	0.586	0.974	0.707
	F-Measure	NaN	0.977	0.923	0.97	0.005	0.97	NaN	0.98	0.737	0.968	0.827
DontKnow	Precision	0.015	0.1	NaN	0.0	0.015	0.037	0.015	0.0	0.022	0.0	0.041
	Recall	1.0	0.167	0.0	0.0	1.0	0.167	1.0	0.0	0.667	0.0	0.833
	F-Measure	0.029	0.125	NaN	NaN	0.029	0.061	0.029	NaN	0.043	NaN	0.078
Weighted Avg	Precision	NaN	0.959	NaN	NaN	0.986	0.942	NaN	0.962	0.963	0.937	0.974
	Recall	0.015	0.952	0.855	0.939	0.019	0.913	0.015	0.93	0.567	0.93	0.707
	F-Measure	NaN	0.955	NaN	NaN	0.01	0.919	NaN	NaN	0.703	NaN	0.812
	ρ-Accuracy	0.03	**0.957**	0.759	0.941	0.037	0.941	0.03	**0.952**	0.719	0.939	0.818

Table 10.29: TCMU SCALL NPIR TNRR Greedy on Manually Annotated dataset for Location

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	NaN	NaN	NaN	NaN	0.6	0.65	NaN	0.778	0.75	0.5	1.0
	Recall	0.0	0.0	0.0	0.0	0.167	0.722	0.0	0.778	0.167	0.333	0.111
	F-Measure	NaN	NaN	NaN	NaN	0.261	0.684	NaN	0.778	0.273	0.4	0.2
NonMatch	Precision	NaN	0.997	0.944	NaN	0.977	0.972	NaN	0.975	0.993	0.958	0.982
	Recall	0.0	0.807	1.0	0.0	0.967	0.979	0.0	0.99	0.728	0.985	0.974
	F-Measure	NaN	0.892	0.971	NaN	0.972	0.976	NaN	0.982	0.84	0.971	0.978
DontKnow	Precision	0.015	0.051	0.0	0.015	0.087	0.0	0.015	NaN	0.032	0.0	0.0
	Recall	1.0	0.833	0.0	1.0	0.333	0.0	1.0	0.0	0.667	0.0	0.0
	F-Measure	0.029	0.096	NaN	0.029	0.138	NaN	0.029	NaN	0.062	NaN	NaN
Weighted Avg	Precision	NaN	NaN	NaN	NaN	0.947	0.944	NaN	NaN	0.968	0.924	0.968
	Recall	0.015	0.772	0.942	0.015	0.923	0.954	0.015	0.966	0.702	0.942	0.923
	F-Measure	NaN	NaN	NaN	NaN	0.928	NaN	NaN	NaN	0.804	NaN	NaN
	ρ-Accuracy	0.03	0.871	0.943	0.03	0.941	0.939	0.03	**0.956**	0.82	0.929	**0.951**

Table 10.30: TCBI SCALL NPIR TNCC Knowledge on Manually Annotated dataset for Location

Considering a Character-based Relative Completeness comparison approach (Char-RC), and rules extracted relying on binary classification (TCBI) considering all the sources at the same time (SCALL), the configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistency removal normalization (NPIR), independently from the threshold normalization function, using Jaro-Winkler similarity metric. The detailed results are presented in table 10.32. It is interesting to see how also QGram and Taglink similarity metrics performed well. It is important to notice that the Jaro-Winkler did not support very precise positive matching decisions, although the recall was the best one. In this case, the best precision was achieved by Taglink, Monge-Elkan and Levenshtein similarity metrics. However, Levenshtein and Monge-Elkan did not produce the best recall of negative matching pairs.

If we consider the same settings, but learn rules considering also the third "don't know" class (TCMU), the configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistency removal normalization (NPIR), applying conservative thresholds on atoms of positive matching and relaxed threshold on negative matching rules (TCRR), using QGram similarity metric. Also in this case, considering multiclass classification, rules learned and applied using Taglink reduced performances, whereas QGram allowed to learn and apply rule with a perfect precision

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	NaN	0.889	NaN	NaN	0.6	1.0	NaN	1.0	0.75	0.667	0.812
	Recall	0.0	0.444	0.0	0.0	0.167	0.222	0.0	0.167	0.167	0.111	0.722
	F-Measure	NaN	0.593	NaN	NaN	0.261	0.364	NaN	0.286	0.273	0.19	0.765
NonMatch	Precision	NaN	0.963	0.944	0.944	0.993	NaN	NaN	0.983	0.991	0.945	0.975
	Recall	0.0	0.995	1.0	0.997	0.733	0.0	0.0	0.874	0.884	0.797	0.992
	F-Measure	NaN	0.979	0.971	0.97	0.843	NaN	NaN	0.925	0.935	0.865	0.983
DontKnow	Precision	0.015	0.5	0.0	0.0	0.033	0.015	0.015	0.062	0.081	0.024	0.0
	Recall	1.0	0.167	0.0	0.0	0.667	1.0	1.0	0.667	0.833	0.333	0.0
	F-Measure	0.029	0.25	NaN	NaN	0.063	0.029	0.029	0.114	0.147	0.045	NaN
Weighted Avg	Precision	NaN	0.953	NaN	NaN	0.962	NaN	NaN	0.97	0.968	0.92	0.954
	Recall	0.015	0.959	0.942	0.939	0.707	0.024	0.015	0.84	0.852	0.76	0.966
	F-Measure	NaN	0.951	NaN	NaN	0.806	NaN	NaN	0.886	0.894	0.823	NaN
	ρ-Accuracy	0.03	**0.959**	0.943	0.941	0.819	0.049	0.03	0.906	0.914	0.84	**0.961**

Table 10.31: TCMU SCALL NPIR TNRR Knowledge on Manually Annotated dataset for Location

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.21	1.0	NaN	0.206	0.8	1.0	NaN	0.714	NaN	0.333	1.0
	Recall	0.722	0.111	0.0	0.722	0.222	0.056	0.0	0.833	0.0	0.056	0.111
	F-Measure	0.325	0.2	NaN	0.321	0.348	0.105	NaN	0.769	NaN	0.095	0.2
NonMatch	Precision	NaN	0.985	1.0	0.977	0.965	0.98	NaN	0.979	0.984	0.948	0.987
	Recall	0.0	0.833	0.026	0.874	0.985	0.879	0.0	0.972	0.928	0.987	0.943
	F-Measure	NaN	0.903	0.05	0.923	0.975	0.927	NaN	0.975	0.955	0.967	0.965
DontKnow	Precision	0.011	0.024	0.015	0.5	0.0	0.048	0.015	0.0	0.022	0.2	0.077
	Recall	0.667	0.333	1.0	0.167	0.0	0.5	1.0	0.0	0.167	0.167	0.5
	F-Measure	0.022	0.045	0.029	0.25	NaN	0.087	0.029	NaN	0.038	0.182	0.133
Weighted Avg	Precision	NaN	0.972	NaN	0.936	0.944	0.967	NaN	0.953	NaN	0.91	0.974
	Recall	0.041	0.794	0.039	0.857	0.937	0.838	0.015	0.952	0.877	0.935	0.901
	F-Measure	NaN	0.859	NaN	0.887	NaN	0.879	NaN	NaN	NaN	0.918	0.919
	ρ-Accuracy	0.059	0.879	0.075	0.766	**0.948**	0.903	0.03	**0.944**	0.926	0.934	**0.942**

Table 10.32: TCBI SCALL NPIR TNCC Char-RC on Manually Annotated dataset for Location

in matching, and also the third best (surprisingly equal similarity produced a high recall). Also rules learned and applied using Levenshtein similarity metric produced good results, with a lower precision with respect to QGram supporting matching decision, but with a higher recall.

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.21	0.833	NaN	NaN	0.611	1.0	NaN	1.0	NaN	0.167	0.667
	Recall	0.722	0.278	0.0	0.0	0.611	0.111	0.0	0.056	0.0	0.056	0.444
	F-Measure	0.325	0.417	NaN	NaN	0.611	0.2	NaN	0.105	NaN	0.083	0.533
NonMatch	Precision	NaN	0.986	0.933	0.966	0.983	1.0	NaN	0.982	0.984	0.946	0.989
	Recall	0.0	0.913	0.72	0.879	0.892	0.003	0.0	0.956	0.923	0.985	0.907
	F-Measure	NaN	0.948	0.813	0.921	0.935	0.005	NaN	0.969	0.952	0.965	0.946
DontKnow	Precision	0.011	0.043	0.018	0.034	0.071	0.015	0.015	0.061	0.021	0.5	0.045
	Recall	0.667	0.333	0.333	0.333	0.5	1.0	1.0	0.333	0.167	0.167	0.333
	F-Measure	0.022	0.075	0.034	0.062	0.125	0.029	0.029	0.103	0.037	0.25	0.08
Weighted Avg	Precision	NaN	0.966	NaN	NaN	0.954	0.986	NaN	0.969	NaN	0.905	0.961
	Recall	0.041	0.877	0.683	0.833	0.874	0.022	0.015	0.908	0.872	0.932	0.879
	F-Measure	NaN	0.912	NaN	NaN	0.909	0.014	NaN	0.919	NaN	0.916	0.916
	ρ-Accuracy	0.059	**0.926**	0.789	0.895	0.901	0.043	0.03	**0.943**	0.924	0.923	0.916

Table 10.33: TCMU SCALL NPIR TNCR Char-RC on Manually Annotated dataset for Location

Performing experiments with the New York Times dataset performances decrease in terms of recall as shown in table 10.34. Considering rules extracted relying on binary classification (TCBI) considering all the sources at the same time (SCALL), the configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistency operator normalization (NPIC), and Conservative Match and Non Match threshold normalization (TNCC), using Jaccard similarity metric and relying on comparison method based on Character-based Relative Completeness comparison method. An explanation to these results can be given considering the low number of positive matching examples in the training set, not supporting the definition of permissive thresholds for rules satisfaction. However, the precision of matching samples is perfect in all considered cases. Character-based Relative Completeness weights the score of similarity to the longest variant of the attribute value. This type of measure tends to decrease the similarity score of attributes, increasing precision. However, in this case it allowed to normalize matching the different variants of the attributes to allow defined rules to be satisfied by a larger number of sample pairs. The second best result is obtained using Taglink and a greedy comparison method. Notice that in this case, the operator inconsistency normalization process that performed better is the inconsistency normalization. This approach tends to build more conservative rules, as rather than removing atoms switches the operator to make it consistent with the rule target. Longer rules are less likely to be satisfied as more attributes need to be involved in the process. However, in this case, the definition of the thresholds values learned according to a normalized similarity score seems to have higher impact. Notice that the ρ-accuracy has the same value as the F-measure in the best case. This implies that no false negative matching decisions were taken, and thus when no positive matching decision could be taken, no negative matching rule was satisfied.

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Greedy	Precision	NaN	1.0	1.0	NaN	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	Recall	0.0	0.034	0.324	0.0	0.023	0.012	0.042	0.003	0.221	0.002	0.34
	F-Measure	NaN	0.066	0.49	NaN	0.044	0.024	0.08	0.006	0.362	0.005	0.507
	ρ-Accuracy	0.0	0.066	0.489	0.0	0.044	0.024	0.073	0.006	0.362	0.004	**0.506**
Knowledge	Precision	NaN	1.0	1.0	NaN	1.0	1.0	NaN	1.0	1.0	1.0	1.0
	Recall	0.0	0.096	0.016	0.0	0.19	0.103	0.0	0.369	0.037	0.014	0.023
	F-Measure	NaN	0.175	0.031	NaN	0.319	0.187	NaN	0.539	0.072	0.028	0.045
	ρ-Accuracy	0.0	0.175	0.029	0.0	0.315	0.181	0.0	**0.483**	0.071	0.024	0.039
Char-RC	Precision	1.0	1.0	1.0	1.0	1.0	1.0	NaN	1.0	1.0	1.0	1.0
	Recall	0.363	0.01	0.127	0.601	0.106	0.132	0.0	0.017	0.018	0.003	0.001
	F-Measure	0.533	0.019	0.225	0.751	0.191	0.233	NaN	0.034	0.036	0.005	0.003
	ρ-Accuracy	0.533	0.02	0.215	**0.751**	0.185	0.225	0.0	0.033	0.035	0.006	0.002

Table 10.34: TCBI SCALL NPIC TNCC on NYT dataset for Location

If we consider the same settings, but learn rules considering also the third “don’t know” class (TCMU), the configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistent operator removal normalization (NPIR), and Conservative Match and Non Match threshold normalization (TNCC), using Jaro-Winkler similarity metric and relying on comparison method based on Character-based Relative Completeness comparison method (table 10.35). The second best performance using multiclass classification and relative completeness similarity score normalization was obtained relying on Taglink similarity metric. The greedy approach on the matching functional approach seems to heavily affect the matching recall of locations.

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Greedy	Precision	NaN	1.0	1.0	1.0	1.0	1.0	NaN	1.0	1.0	1.0	1.0
	Recall	0.0	0.066	0.324	0.008	0.004	0.041	0.0	0.019	0.056	0.006	0.097
	F-Measure	NaN	0.124	0.49	0.017	0.008	0.079	NaN	0.037	0.106	0.012	0.177
	ρ-Accuracy	0.0	0.107	0.489	0.013	0.008	0.079	0.0	0.037	0.095	0.012	0.177
Knowledge	Precision	NaN	1.0	1.0	1.0	1.0	1.0	NaN	1.0	1.0	1.0	1.0
	Recall	0.0	0.02	0.009	0.01	0.19	0.18	0.0	0.006	0.019	0.049	0.031
	F-Measure	NaN	0.04	0.017	0.02	0.319	0.305	NaN	0.012	0.038	0.094	0.061
	ρ-Accuracy	0.0	0.035	0.016	0.017	0.316	0.305	0.0	0.012	0.037	0.086	0.05
Char-RC	Precision	1.0	1.0	1.0	1.0	1.0	1.0	NaN	1.0	1.0	1.0	1.0
	Recall	0.363	0.162	0.033	0.011	0.526	0.186	0.0	0.133	0.085	0.123	0.318
	F-Measure	0.533	0.278	0.063	0.021	0.689	0.314	NaN	0.235	0.157	0.22	0.483
	ρ-Accuracy	0.533	0.226	0.05	0.014	**0.684**	0.313	0.0	0.218	0.156	0.135	0.471

Table 10.35: TCMU SCALL NPIC TNCC on NYT dataset for Location

A consideration related to this experiment is that with the NYT dataset, no similarity metric seems to steadily the best option relying on different learning and comparison methods. More considerations also on the nature of the set considered will be presented at the end of the section.

Bottom-up Rules for Organization

In this section we present the experiments that obtained the best results on organizations evaluation datasets. In particular, for each comparison method, and classification method, we select the best inconsistency normalization and threshold normalization combination based on the ρ-accuracy described in equation (10.1) at the beginning of the chapter. The detailed results of the other experiments will be available online, together with the raw experiment results and the datasets.

Considering a greedy comparison approach (Greedy), and rules extracted relying on binary classification (TCBI) considering all the sources at the same time (SCALL), the configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistency removal, independently from the threshold normalization function, and relying on the Levenshtein similarity metric (table 10.36). Also Jaccard and Jaro-Winkler string similarity metrics performed very similarity with a higher positive matching precision. The quality of the positive matching performances is rather low, whereas negative matching classification was done quite accurately. The overall accuracy of the experiment is good, but the poor positive matching performances seem to suggest that the training is to poor in terms of positive matching samples.

Considering a Knowledge-driven comparison approach (Knowledge), and rules extracted relying on binary classification (TCBI) considering all the sources at the same time (SCALL), the configuration that produced the best results in terms of ρ-accuracy

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.2	0.2	0.118	0.3	0.5	NaN	1.0	0.5	NaN	0.304	0.583
	Recall	0.029	0.029	0.059	0.088	0.176	0.0	0.088	0.235	0.0	0.706	0.206
	F-Measure	0.051	0.051	0.078	0.136	0.261	NaN	0.162	0.32	NaN	0.425	0.304
NonMatch	Precision	0.707	0.978	0.981	0.98	0.983	NaN	0.984	0.982	NaN	0.982	1.0
	Recall	0.061	0.977	0.969	0.968	0.963	0.0	0.891	0.971	0.0	0.201	0.131
	F-Measure	0.113	0.978	0.975	0.974	0.973	NaN	0.935	0.976	NaN	0.333	0.232
DontKnow	Precision	0.012	0.161	0.153	0.136	0.147	0.018	0.066	0.172	0.018	0.013	0.021
	Recall	0.6	0.36	0.36	0.36	0.44	1.0	0.48	0.4	1.0	0.56	1.0
	F-Measure	0.023	0.222	0.214	0.198	0.22	0.035	0.116	0.241	0.035	0.026	0.04
Weighted Avg	Precision	0.682	0.945	0.945	0.949	0.957	NaN	0.968	0.956	NaN	0.948	0.972
	Recall	0.07	0.943	0.936	0.936	0.934	0.018	0.864	0.943	0.018	0.219	0.149
	F-Measure	0.109	0.942	0.94	0.94	0.942	NaN	0.902	0.947	NaN	0.33	0.23
	ρ-Accuracy	0.129	**0.955**	0.942	0.95	0.952	0.035	0.921	0.953	0.035	0.327	0.256

Table 10.36: TCBI SCALL NPIR TNCC Greedy on Manually Annotated dataset for Organization

is the one that applied inconsistency removal normalization (NPIR), independently from the threshold normalization function, using Levenshtein similarity metric. The detailed results are presented in table 10.37. Also Jaro-Winkler, Jaccard and QGram performed well.

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.2	0.2	0.118	0.2	0.5	NaN	1.0	0.5	1.0	0.304	0.538
	Recall	0.029	0.029	0.059	0.029	0.176	0.0	0.088	0.235	0.206	0.706	0.206
	F-Measure	0.051	0.051	0.078	0.051	0.261	NaN	0.162	0.32	0.341	0.425	0.298
NonMatch	Precision	0.87	0.978	0.981	0.976	0.983	NaN	0.984	0.982	0.989	0.982	1.0
	Recall	0.175	0.977	0.969	0.975	0.963	0.0	0.891	0.971	0.871	0.241	0.131
	F-Measure	0.292	0.978	0.975	0.976	0.973	NaN	0.935	0.976	0.926	0.387	0.232
DontKnow	Precision	0.014	0.161	0.153	0.127	0.147	0.018	0.066	0.172	0.071	0.013	0.02
	Recall	0.64	0.36	0.36	0.28	0.44	1.0	0.48	0.4	0.6	0.52	0.96
	F-Measure	0.028	0.222	0.214	0.175	0.22	0.035	0.116	0.241	0.127	0.026	0.039
Weighted Avg	Precision	0.839	0.945	0.945	0.942	0.957	NaN	0.968	0.956	0.973	0.948	0.971
	Recall	0.18	0.943	0.936	0.94	0.934	0.018	0.864	0.943	0.85	0.257	0.148
	F-Measure	0.281	0.942	0.94	0.939	0.942	NaN	0.902	0.947	0.898	0.381	0.23
	ρ-Accuracy	0.297	*0.955*	0.942	0.953	0.952	0.035	0.921	0.953	0.914	0.373	0.255

Table 10.37: TCBI SCALL NPIR Knowledge on Manually Annotated dataset for Organization

Considering a Character-based Relative Completeness comparison approach (Char-RC), and rules extracted relying on binary classification (TCBI) considering all the sources at the same time (SCALL), the configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistency removal normalization (NPIR), independently from the threshold normalization function, using Needlman-Wunsch similarity metric. The detailed results are presented in table 10.38. It is interesting to see how also Jaro-Winkler, Jaccard and Levenshtein similarity metrics performed well. Also in this case, positive matching performances are very poor.

If we consider the same settings, but learn rules considering also the third "don't know" class (TCMU) and a greedy approach, the configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistent operator removal normalization (NPIR), independently from the threshold normalization, using Needlman-Wunsch similarity metric (table 10.39). The second best performance using

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.2	0.2	0.121	0.2	0.5	NaN	1.0	0.4	1.0	0.2	0.583
	Recall	0.029	0.029	0.118	0.029	0.176	0.0	0.088	0.176	0.206	0.029	0.206
	F-Measure	0.051	0.051	0.119	0.051	0.261	NaN	0.162	0.245	0.341	0.051	0.304
NonMatch	Precision	0.707	0.978	0.98	0.976	0.983	NaN	0.984	0.983	0.988	0.978	0.995
	Recall	0.061	0.977	0.969	0.975	0.969	0.0	0.89	0.957	0.858	0.978	0.137
	F-Measure	0.113	0.978	0.974	0.976	0.976	NaN	0.935	0.97	0.919	0.978	0.241
DontKnow	Precision	0.012	0.161	0.167	0.127	0.152	0.018	0.066	0.125	0.061	0.164	0.02
	Recall	0.6	0.36	0.28	0.28	0.4	1.0	0.48	0.4	0.56	0.36	0.96
	F-Measure	0.023	0.222	0.209	0.175	0.22	0.035	0.115	0.19	0.111	0.225	0.039
Weighted Avg	Precision	0.682	0.945	0.945	0.942	0.956	NaN	0.968	0.954	0.972	0.945	0.967
	Recall	0.07	0.943	0.936	0.94	0.939	0.018	0.864	0.928	0.837	0.944	0.154
	F-Measure	0.109	0.942	0.94	0.939	0.945	NaN	0.901	0.938	0.89	0.942	0.239
	ρ-Accuracy	0.129	0.955	0.927	0.953	0.955	0.035	0.92	0.946	0.906	**0.956**	0.262

Table 10.38: TCBI SCALL NPIR Char-RC on Manually Annotated dataset for Organization

multiclass classification and relative completeness similarity score normalization was obtained relying on Levenshtein similarity metric. Also in this case, positive matching performances are quite poor. Considering Knowledge-driven approach, the performances do not change, therefore we avoid presenting them twice.

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.2	0.2	0.105	0.296	0.083	NaN	NaN	0.417	NaN	0.2	0.121
	Recall	0.029	0.029	0.118	0.706	0.059	0.0	0.0	0.147	0.0	0.029	0.118
	F-Measure	0.051	0.051	0.111	0.417	0.069	NaN	NaN	0.217	NaN	0.051	0.119
NonMatch	Precision	0.707	0.978	0.982	0.983	0.981	0.981	NaN	0.985	0.991	0.978	0.989
	Recall	0.061	0.977	0.959	0.173	0.966	0.194	0.0	0.949	0.844	0.978	0.943
	F-Measure	0.113	0.978	0.971	0.294	0.973	0.324	NaN	0.966	0.912	0.978	0.966
DontKnow	Precision	0.012	0.161	0.189	0.014	0.196	0.02	0.018	0.125	0.062	0.164	0.136
	Recall	0.6	0.36	0.4	0.6	0.44	0.92	1.0	0.48	0.64	0.36	0.48
	F-Measure	0.023	0.222	0.256	0.027	0.272	0.04	0.035	0.198	0.113	0.225	0.212
Weighted Avg	Precision	0.682	0.945	0.947	0.949	0.945	NaN	NaN	0.955	NaN	0.945	0.953
	Recall	0.07	0.943	0.929	0.194	0.934	0.202	0.018	0.921	0.82	0.944	0.915
	F-Measure	0.109	0.942	0.937	0.292	0.939	NaN	NaN	0.934	NaN	0.942	0.932
	ρ-Accuracy	0.129	0.955	0.921	0.294	0.934	0.335	0.035	0.945	0.898	**0.956**	0.921

Table 10.39: TCMU SCALL IPP NPIR Greedy Manually Annotated Dataset for organization

If we consider also the third “don’t know” class (TCMU) and a Character-based Relative Completeness weighted approach, the configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistent operator removal normalization (NPIR), independently from the threshold normalization, using Jaro-Winkler similarity metric (table 10.40). Also in this case, Needleman-Wunsch similarity metric allowed extracting rules performing fairly well. Matching performances for positive matching samples are quite poor, reflecting weakness of the considered training set.

Performing entity matching experiments on the NYT dataset using matching rules that are the result of the bottom up process described in section 8.2 produced the results presented in tables 10.41 and 10.42. The configuration that produced the best results in terms of ρ-accuracy is the one that applied inconsistent operator removal normalization (NPIR), independently from the threshold normalization, using

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	0.2	0.2	0.111	0.2	0.333	NaN	NaN	0.455	NaN	0.2	0.5
	Recall	0.029	0.029	0.059	0.029	0.029	0.0	0.0	0.147	0.0	0.029	0.147
	F-Measure	0.051	0.051	0.077	0.051	0.054	NaN	NaN	0.222	NaN	0.051	0.227
NonMatch	Precision	0.707	0.98	0.982	0.997	0.981	0.982	NaN	0.983	NaN	0.978	0.992
	Recall	0.061	0.953	0.279	0.269	0.978	0.867	0.0	0.953	0.0	0.978	0.93
	F-Measure	0.113	0.966	0.434	0.424	0.979	0.921	NaN	0.968	NaN	0.978	0.96
DontKnow	Precision	0.012	0.11	0.022	0.024	0.186	0.055	0.018	0.112	0.018	0.164	0.12
	Recall	0.6	0.4	0.88	1.0	0.44	0.48	1.0	0.4	1.0	0.36	0.64
	F-Measure	0.023	0.172	0.043	0.047	0.262	0.099	0.035	0.175	0.035	0.225	0.203
Weighted Avg	Precision	0.682	0.946	0.943	0.961	0.951	NaN	NaN	0.955	NaN	0.945	0.965
	Recall	0.07	0.921	0.284	0.276	0.946	0.839	0.018	0.924	0.018	0.944	0.906
	F-Measure	0.109	0.93	0.419	0.408	0.944	NaN	NaN	0.936	NaN	0.942	0.929
	ρ-Accuracy	0.129	0.945	0.43	0.43	**0.961**	0.905	0.035	0.947	0.035	0.956	0.941

Table 10.40: TCMU SCALL NPIR TNCC Char-RC Manually Annotated Dataset for Organization

Needlaman-Wunsch similarity metric and relying on the Knowledge-driven comparison method and binary classification method (TCBI). The greedy comparison method, also relying on Needlaman-Wunsch similarity metric produced very similar results in terms of accuracy. None of the other considered metrics produced any decent results. Considering multiclass classification learning method (TCMU) the configuration that performed better did not change, but in the similarity metric that obtained positive results which in this case is Jaccard. Differently from the case of locations, multiclass classification produced better results in terms of accuracy than binary classification.

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Greedy	Precision	1.0	NaN	1.0	1.0	1.0	NaN	1.0	1.0	NaN	1.0	1.0
	Recall	0.001	0.0	0.039	0.038	0.024	0.0	0.045	0.039	0.0	0.544	0.04
	F-Measure	0.002	NaN	0.075	0.073	0.048	NaN	0.086	0.075	NaN	0.705	0.077
	ρ-Accuracy	0.001	0.007	0.061	0.06	0.042	0.0	0.078	0.061	0.0	0.688	0.077
Knowledge	Precision	1.0	NaN	1.0	NaN	1.0	NaN	1.0	1.0	1.0	1.0	1.0
	Recall	0.001	0.0	0.038	0.0	0.026	0.0	0.048	0.041	0.059	0.544	0.043
	F-Measure	0.002	NaN	0.073	NaN	0.05	NaN	0.091	0.08	0.111	0.705	0.082
	ρ-Accuracy	0.001	0.007	0.06	0.007	0.043	0.0	0.081	0.064	0.103	**0.69**	0.081
Char-RC	Precision	1.0	1.0	1.0	NaN	1.0	NaN	1.0	1.0	1.0	NaN	1.0
	Recall	0.001	0.004	0.046	0.0	0.024	0.0	0.044	0.037	0.059	0.0	0.039
	F-Measure	0.002	0.007	0.089	NaN	0.048	NaN	0.084	0.071	0.111	NaN	0.075
	ρ-Accuracy	0.001	0.013	0.069	0.007	0.041	0.0	0.077	0.058	0.102	0.007	0.075

Table 10.41: TCBI SCALL IPP NPIR TNCC NYT Dataset for Organization

		EQ.	Lev.	Euc.	Jac.	Jar.	Mgk.	Ovl.	QGr.	SMW.	NWu.	Tag.
Match	Precision	1.0	1.0	1.0	1.0	1.0	NaN	NaN	1.0	NaN	1.0	1.0
	Recall	0.001	0.004	0.048	0.548	0.028	0.0	0.0	0.004	0.0	0.001	0.035
	F-Measure	0.002	0.007	0.091	0.708	0.055	NaN	NaN	0.007	NaN	0.002	0.068
	ρ-Accuracy	0.001	0.013	0.071	0.694	0.047	0.004	0.0	0.013	0.009	0.009	0.061
Match	Precision	1.0	1.0	1.0	1.0	1.0	NaN	NaN	1.0	NaN	1.0	1.0
	Recall	0.001	0.005	0.048	0.548	0.026	0.0	0.0	0.005	0.0	0.001	0.005
	F-Measure	0.002	0.01	0.091	0.708	0.05	NaN	NaN	0.01	NaN	0.002	0.01
	ρ-Accuracy	0.001	0.014	0.091	**0.695**	0.043	0.004	0.0	0.015	0.009	0.009	0.01
Match	Precision	1.0	1.0	1.0	1.0	NaN	NaN	NaN	1.0	NaN	NaN	1.0
	Recall	0.001	0.004	0.037	0.002	0.0	0.0	0.0	0.004	0.0	0.0	0.004
	F-Measure	0.002	0.007	0.071	0.005	NaN	NaN	NaN	0.007	NaN	NaN	0.007
	ρ-Accuracy	0.001	0.013	0.071	0.008	0.007	0.008	0.0	0.013	0.0	0.007	0.014

Table 10.42: TCMU SCALL IPP NPIR NYT Dataset for Organization

10.2.3 Mixed Rules Experiments

In this section we aim at presenting the results of mixing the entity matching rules result of top-down and bottom up processes. Namely, we aim at joining and mixing the set of rules learned in a bottom up fashion together with the set of rules result of ontological analysis. The goal of these experiments is to understand whether different approaches in mixing the rules can proving any improvement with respect to the simple application of just bottom up or top down rules. Therefore, in the following we'll present the best results obtained mixing the rules for each of the datasets considered, and confront them with the best results of the application of the single approach. This time, we will not present in tables the results obtained for each comparison method, but we rather present the details related to the best configuration.

It is important to notice that the selection of the best configuration is done in a pool of thousands of experiment results which are not worthy present in detail in this context, but that will be available online, together with raw experiment results and dataset used.

Mixing Rules for Person

	TP	FP	Prec	Rec	F-2
M	0.026	0.001	0.945	0.397	0.559
N	0.817	0.035	0.959	0.915	0.937
U	0.019	0.102	0.158	0.443	0.233
Avg	0.731	0.036	0.923	0.861	0.882

Table 10.43: IPTO, Knowledge, Needleman-Wunsch, Mixed rules

	TP	FP	Prec	Rec	F-2
M	0.026	0.001	0.945	0.397	0.559
N	0.818	0.037	0.957	0.917	0.936
U	0.019	0.099	0.158	0.432	0.232
Avg	0.732	0.037	0.922	0.862	0.882

Table 10.44: IPP, KNOWLEDGE, Needleman-Wunsch, Mixed rules

	TP	FP	Prec	Rec	F-2
M	0.026	0.002	0.929	0.397	0.556
N	0.816	0.034	0.96	0.915	0.937
U	0.019	0.103	0.157	0.443	0.232
Avg	0.731	0.035	0.923	0.861	0.882

Table 10.45: IPTO, KNOWLEDGE, Needleman-Wunsch, Bottom up only

	TP	FP	Prec	Rec	F-2
M	0.0	0.0	1.0	0.008	0.015
N	0.872	0.068	0.927	0.977	0.952
U	0.014	0.045	0.233	0.318	0.269
Avg	0.779	0.063	0.902	0.886	0.862

Table 10.46: KNOWLEDGE, Needleman-Wunsch, Diachronic only Top-down

	TP	FP	Prec	Rec	F-2
M	0.019	0.001	0.929	0.298	0.451
N	0.872	0.057	0.939	0.977	0.958
U	0.013	0.038	0.252	0.295	0.272
Avg	0.78	0.052	0.909	0.904	0.896

Table 10.47: KNOWLEDGE, Needleman-Wunsch, Combinatorial only Top-down

	TP	FP	Prec	Rec	F-2
M	0.025	0.001	0.943	0.382	0.543
N	0.817	0.035	0.959	0.915	0.937
D	0.019	0.103	0.157	0.443	0.231
Avg	0.731	0.036	0.923	0.86	0.881

Table 10.48: IPP, KNOWLEDGE, Needleman -Wunsch, Mixing with Combinatorial Top down

Table 10.49: Experiments with mixed rules on Manually Annotated dataset for Person

Considering the manually annotated dataset, the configuration that performed better in terms of ρ-accuracy is:

- Knowledge-driven comparison method, with Needleman-Wunsch similarity metric, applying inconsistency removal normalization (NPIR), threshold normalization conservative for positive matching rules and relaxed for negative matching rules (TNCR), integrating only positive matching top-down rules (IPTO) (table 10.43). The ρ-accuracy is 0.907. If we consider plain rules integration, the ρ-accuracy is the same, but the precision of negative match decreases, despite an increase of recall (table 10.44).

If we compare the results obtained with mixed rules with results obtained by applying only bottom-up or top-down rules, we can find that mixed rules performed better that bottom up rules in terms of precision of positive and negative match classification. The improvement is minimal, but not irrelevant (table 10.45). If we consider the comparison of matching with diachronic top-down only matching results, the improvements in terms of precision of negative matching precision are small but not irrelevant, whereas the improvements in terms of recall are quite considerable for positive matching classification (table 10.46). This shows how learning matching thresholds can improve matching performances with respect to real world data. The rule learning method proposed in this context is not at the state of the art in terms of regression method, but allowed us to test the integrated rules combination. Comparing the results of mixed rules combination considering only diachronic top-down rules with the results of top-down rules obtained through combination of functional attributes defined in the identification ontology, we can see how the second performed better in classifying negative matching rules, specially in terms of precision. This is due to the fact that negative matching rules defined combining a functional property with the name, allows to reduce the effect of problems related to string comparison (table 10.47). However, if we test the integration of learned rules, together with the rules obtained through combination of functional attributes, we obtain results similar, even thou slight worst than the one obtained mixing learned rules with diachronic rules (table 10.48).

Considering the NYT Dataset, the configuration we considered the configuration that performed better in terms of ρ-accuracy according to each of the comparison methods:

- considering greedy approach, the configuration that performed better is the one that applied inconsistencies normalization process (NPIC), setting relaxed thresholds for positive and negative matching rules, relying on Levenshtein similarity metric. This result was obtained with positive only top-down rules integration.
- considering knowledge-driven approach, the configuration that performed better is the one that applied inconsistencies normalization process (NPIC), setting conservative thresholds for positive and negative matching rules, and relying on Jaro-

Class	TP	FP	Precision	Recall	F-Measure	ρ-accuracy
Greedy	0.755	0.0	1.0	0.757	0.861	0.86
Knowledge	0.783	0.0	1.0	0.785	0.879	0.879
Char-RC	0.872	0.0	1.0	0.874	0.933	0.902

Table 10.50: NYT Dataset mixed rules, 3 comparison methods for Person

Winkler string similarity metric. This result was obtained with positive only top-down rules integration.

- considering Character-based Relative Completeness method, the configuration that performed better is the one that applied inconsistencies removal process (NPIR), setting relaxed thresholds for positive and negative matching rules, and relying on Monge-Elkan string similarity metric. This result was obtained with positive only top-down rules integration.

The configuration that performed better in absolute terms is the one that relied on Character-based Relative Completeness, with a ρ-accuracy of 0.902. However, analyzing the results obtained in table 10.50, we can notice that the ρ-accuracy is 3 points lower than the F-measure. This implies that Char-RC comparison method allowed to learn more permissive threshold allowing to capture a large number of match, but also supported a large number of false negative classification. The other two comparison approaches had basically no variations considering F-measure and ρ-accuracy, and thus, they can be considered more reliable. Analyzing their configuration, the reason of this higher reliability is in the inconsistent atoms normalization process. In fact, the inconsistency normalization NPIC reverses the sign of the operator when inconsistent with the matching rule, generating more conservative rules. Given these consideration, we consider as the best the method that relied on Knowledge-driven approach for comparison.

Comparing the results running experiments on the NYT datasets, we can notice that the application of a mixed approach performed a little worst than the pure bottom-up approach as shown in table 10.51. It seems that the integration of Diachronic only top-down rules decreased the matching performances of the bottom-up learned rules. This can be due to the fact that Diachronic rules imposed a more conservative threshold on some positive matching rules, reducing the recall of less than a point (Mixed-D in table 10.51). The performances are not even comparable when considering only matching with diachronic rules (Top-D in table 10.51). This may be a sign that inverse-functional properties in the dataset have are rare and have a very low similarity. This explains also better performances of Char-RC classifier, as, in a sense, relaxed matching

Class	TP	FP	Precision	Recall	F-Measure	ρ-accuracy
Mixed-D	0.783	0.0	1.0	0.785	0.879	0.879
Bottom-up	0.79	0.0	1.0	0.791	0.884	0.882
Top-D	0.002	0.875	0.002	1.0	0.004	0.002
Top-C	0.589	0.0	1.0	0.591	0.743	0.604
Mixed-C	0.846	0.0	1.0	0.848	0.918	0.916

Table 10.51: NYT Dataset, comparing approaches for Person

requirements thresholds based on variations of attributes values. A more in the depth analysis on the characteristics of the matched dataset is going to be presented in section 10.2.4. Considering rules built relying on combination of attributes defined in the identification ontology (Top-C in table 10.51), the results obtained are almost 27 points worst than the one obtained relying on a mixed approach, showing another time that learning matching thresholds provide important improvements also in terms of "false negative" classification. In fact, the threshold used for combinatorial negative rules was set to 0.5, showing to be too relaxed. This case may also indicate that the syntactic difference between matching attributes is particularly relevant in this dataset. If we consider mixed rules that is the result of a plain integration (IPP) of bottom up learned rules and combinatorial top-down rules (Mixed-C in table 10.51, we obtain the best classification results with a higher recall and also marginal number of negatively classified rules as shown by the low difference between F-measure and ρ-accuracy.

Comparing the results running experiments on the OAEI 2010 dataset, we considered the configuration that performed better in terms of ρ-accuracy according to each of the comparison methods:

- considering greedy approach, the configuration that performed better is the one that applied inconsistencies normalization process (NPIC), setting conservative thresholds for positive and negative matching rules, relying on Jaro-Winkler similarity metric. This result was obtained with positive only top-down rules integration.
- considering knowledge-driven approach, the configuration that performed better is the one that applied inconsistencies normalization process (NPIC), relaxed threshold for positive matching rules and conservative threshold for negative matching rules, and relying on Needlman-Wunsch string similarity metric. This result was obtained with positive only top-down rules integration.
- considering Character-based Relative Completeness method, the configuration that performed better is the one that applied inconsistencies removal process

Class	TP	FP	Precision	Recall	F-Measure	ρ-accuracy
Greedy	0.976	0.0	1.0	0.976	0.988	0.988
Knowledge	0.972	0.0	1.0	0.972	0.986	0.986
Char-RC	0.986	0.0	1.0	0.986	0.993	0.991

Table 10.52: OAEI 2010 Dataset mixed rules, 3 comparison methods for Person

(NPIR), setting relaxed thresholds for positive and conservative threshold for negative matching rules, and relying on QGram string similarity metric. This result was obtained with positive only top-down rules integration.

The configuration that performed better in absolute terms is the one that relied on Character-based Relative Completeness, with a ρ-accuracy of 0.991. However, analyzing the results obtained in table 10.52, we can notice that the ρ-accuracy is little lower than the F-measure. This implies that Char-RC comparison method allowed to learn more permissive threshold allowing to capture a large number of match, but also supported supported some false negative classification. Furthermore, the configuration that performed better on Char-RC applied inconsistency removal normalization, which creates shorter rules, which are more likely to be satisfied. The other two comparison approaches had basically no variations considering F-measure and ρ-accuracy, and thus, they can be considered more reliable. Analyzing their configuration, the reason of this higher reliability is in the inconsistent atoms normalization process. In fact, the inconsistency normalization NPIC reverses the sign of the operator when inconsistent with the matching rule, generating more conservative rules. Given these consideration, we consider as the best the method that relied on Greedy approach for comparison. Notice that the specific dataset does not present variations of attributes, and therefore the difference between Greedy and Knowledge-driven approach is simply in the definition of little more conservative rules for the second.

Comparing the results running experiments on the OAEI datasets, we can notice that the application of a mixed approach performed performed better than the pure bottom-up approach as shown in table 10.53. This can be due to the fact that Diachronic rules imposed included among the matching rules some inverse-functional properties that were not present in the dataset, increasing the recall of over than 40 points (Mixed-D in table 10.53). this i confirmed when considering only matching with top-down diachronic rules (Top-D in table 10.53). This confirms that inverse-functional properties in the dataset have a strong impact in supporting positive matching decisions. Considering rules built relying on combination of attributes defined in the identification ontology (Top-C in table 10.53), the results obtained are better than the one obtained mixing bottom-up and top-down rules. However, both top-down only

Class	TP	FP	Precision	Recall	F-Measure	ρ-accuracy
Mixed-D	0.976	0.0	1.0	0.976	0.988	0.988
Bottom-up	0.534	0.0	1.0	0.534	0.697	0.674
Top-D	0.954	0.0	1.0	0.954	0.977	0.954
Top-C	0.991	0.0	1.0	0.991	0.996	0.991
Mixed-C	0.633	0.0	1.0	0.633	0.776	0.752

Table 10.53: OAEI 2010 Dataset, comparing approaches for Person

approach present some difference comparing F-measure and ρ-accuracy. This implies that the definition of greedy matching decision based on diachronic attributes may lead to false negative classification. This problem is somehow reduced when considering Top-C rules, as negative matching rules are in a sense more conservative. This case may also indicate that the syntactic difference between matching attributes is particularly relevant in this dataset. If we consider mixed rules that is the result of a plain integration (IPP) of bottom up learned rules and combinatorial top-down rules (Mixed-C in table 10.53, we obtain worst classification performances then considering the mixed rules bottom up and top down diachronic only. This can be explained by the fact that inverse functional diachronic properties cannot be applied as single means to support positive matching decision. In fact, positive matching rules are the result of a combination of at least 4 attributes. Furthermore, top-down combinatorial rules that satisfied many positive cases as shown in Top-C of table 10.53, are merged with bottom-up learned rules, which proved to be more conservative. In addition, conservative threshold normalization arose the bar for positive matching classification, and top-down rules merging imposed more conservative thresholds. Therefore, the effectiveness of applied rules is considerably reduced. This is confirmed by the fact that selecting simply a relaxed threshold normalization for positive matching rules, the matching performances get to a ρ-accuracy of 0.897.

Mixing Rules for Location

Considering the manually annotated dataset, the configurations that performed better in terms of ρ-accuracy this time relied on multiclass classification method, and are:

- Knowledge-driven comparison method, with Levenshtein similarity metric, applying inconsistency removal normalization (NPIR), threshold normalization relaxed for positive matching rules and relaxed for negative matching rules (TNRR), integrating plain positive and negative matching top-down rules (IPP) (table 10.54). The ρ-accuracy is 0.959.

	TP	FP	Prec	Rec	F-2
M	0.019	0.002	0.889	0.444	0.593
N	0.937	0.036	0.963	0.995	0.979
U	0.002	0.002	0.5	0.167	0.25
Avg	0.883	0.034	0.953	0.959	0.951

Table 10.54: IPP, KNOWLEDGE, Levenshtein, Mixed rules

	TP	FP	Prec	Rec	F-2
M	0.029	0.005	0.857	0.667	0.75
N	0.92	0.019	0.979	0.977	0.978
U	0.002	0.024	0.091	0.167	0.118
Avg	0.868	0.019	0.961	0.952	0.956

Table 10.55: IPP, GREEDY, Levenshtein, Mixed rules

	TP	FP	Prec	Rec	F-2
M	0.029	0.005	0.857	0.667	0.75
N	0.92	0.022	0.977	0.977	0.977
U	0.002	0.022	0.1	0.167	0.125
Avg	0.868	0.021	0.959	0.952	0.955

Table 10.56: IPP, GREEDY, Levenshtein, Bottom up only

	TP	FP	Prec	Rec	F-2
M	0.002	0.0	1.0	0.056	0.105
N	0.942	0.056	0.944	1.0	0.971
U	0.0	0.0	NaN	0.0	NaN
Avg	0.887	0.052	NaN	0.944	NaN

Table 10.57: IPTO, KNOWLEDGE, Jaro-Winkler, Top-down only

	TP	FP	Prec	Rec	F-2
M	0.002	0.0	1.0	0.056	0.105
N	0.942	0.056	0.944	1.0	0.971
U	0.0	0.0	NaN	0.0	NaN
Avg	0.887	0.052	NaN	0.944	NaN

Table 10.58: IPTO, KNOWLEDGE, Jaro-Winkler, Combinatorial Top-down only

	TP	FP	Prec	Rec	F-2
M	0.029	0.005	0.857	0.667	0.75
N	0.92	0.022	0.977	0.977	0.977
U	0.002	0.022	0.1	0.167	0.125
Avg	0.868	0.021	0.959	0.952	0.955

Table 10.59: IPTO, KNOWLEDGE, Jaro-Winkler, Combinatorial Top-down mixed

Table 10.60: Experiments for the evaluation of mixed rules on Manually Annotated dataset for Location

- Greedy comparison method, with Levenshtein similarity metric, applying inconsistency removal normalization (NPIR), threshold normalization relaxed for positive matching rules and relaxed for negative matching rules (TNRR), integrating only positive matching top-down rules (IPP) (table 10.55). The ρ-accuracy is 0.959.

Analyzing the results of the best performances for the 2 best comparison methods, the greedy matching approach is the one that performed better. In fact, observing the data in table 10.54 we can see that the positive matching precision had a slightly higher precision, but also a lower recall. Knowledge-driven method classified more precisely 'don't know cases', but the difference between the ρ-accuracy and the f-measure is higher. Therefore, we'll compare results of greedy approach on the application of the mixed rules with the results of the application of rules only bottom-up, top-down, and mixed with combinatorial top-down rules.

Considering performances of bottom-up rules using a greedy approach (table 10.56), we can see that mixing rules did not provide any particular advantage, if not in a small increase of negative matching rules precision. In we compare the results of experiment executed with rules result of diachronic top-down rules only (table 10.57), mixing the rules provided important improvements in terms of both precision and recall of positive matching rules. The application of combinatorial top-down rules did provide any significant change in the performance with respect to simple diachronic rules application (table 10.58), but not with respect to top-down diachronic and bottom-up rules mixed. Also mixing combinatorial top-down rules with bottom-up learned rules did not provide any significant change. This also probably due to the fact that bottom-up rules learned very similar rules to the one obtained through combinations of functional attributes, and therefore the application of relaxed thresholds uniformed the top-down combinatorial to the bottom-up and mixed rules.

Considering the NYT evaluation dataset for location, the configuration that performed better than the other is the greedy approach considering the character-based relative completeness comparison method using binary classification approach, applying inconsistency normalization process (NPIC) and applying conservative thresholds of Jaccard similarity metric. The best results, integrated only positive matching top-down rules. Comparing the results running experiments on the NYT datasets, we can notice that the application of a mixed approach performed better than the pure bottom-up approach as shown in table 10.61. It seems that the integration of positive only diachronic attribute based matching rule allowed to take matching decision on attributes that were not included among the bottom-up rules (Mixed-D in table 10.51). The performances are not even comparable when considering only matching with diachronic rules (Top-D in table 10.51), in fact the thresholds considered hardly allowed to support matching decisions. This may be a sign that inverse-functional properties in the dataset have a very low similarity in terms of Jaccard similarity metric. A more in the depth analysis on the characteristics of the matched dataset is going to be presented in section 10.2.4. Considering rules built relying on combination of attributes defined in the identification ontology (Top-C in table 10.51), the results obtained are not better that top diachronic only, showing another time that learning matching thresholds provide important improvements also in terms of classification. In fact, the threshold used for combinatorial negative rules was set to 0.5, showing to be too relaxed and the threshold for positive match 0.9 was too conservative. This case may also indicate that the syntactic difference between matching attributes is particularly relevant in this dataset. If we consider mixing bottom up learned rules and combinatorial top-down rules (Mixed-C in table 10.61), we obtain a worst classification results with the same precision, recall and f-measure of the bottom-up approach but a larger number of false negative classification, as shown by the difference between the f-measure and the ρ-accuracy.

Mixing Rules for Organization

Considering the manually annotated dataset for entity type organization, there were several configurations that performed equally in terms of ρ-accuracy. The one we decided to analyze and present in this context is:

Class	TP	FP	Precision	Recall	F-Measure	ρ-accuracy
Mixed-D	0.619	0.0	1.0	0.619	0.765	0.765
Bottom-up	0.601	0.0	1.0	0.601	0.751	0.751
Top-D	0.03	0.0	1.0	0.03	0.057	0.032
Top-C	0.03	0.0	1.0	0.03	0.057	0.032
Mixed-C	0.601	0.0	1.0	0.601	0.751	0.612

Table 10.61: NYT Dataset, comparing approaches for Location

	TP	FP	Prec	Rec	F-2
M	0.017	0.039	0.304	0.706	0.425
N	0.504	0.009	0.983	0.526	0.686
U	0.006	0.425	0.013	0.32	0.025
Avg	0.484	0.017	0.949	0.527	0.668

Table 10.62: IPNO, KNOWLEDGE, Needleman-Wunsch, Mixed rules

	TP	FP	Prec	Rec	F-2
M	0.017	0.045	0.276	0.706	0.397
N	0.501	0.009	0.983	0.523	0.683
U	0.006	0.423	0.013	0.32	0.026
Avg	0.48	0.017	0.949	0.524	0.664

Table 10.63: IPP, KNOWLEDGE, Needleman-Wunsch, Mixed rules

	TP	FP	Prec	Rec	F-2
M	0.017	0.039	0.304	0.706	0.425
N	0.231	0.004	0.982	0.241	0.387
U	0.009	0.699	0.013	0.52	0.026
Avg	0.222	0.018	0.948	0.257	0.381

Table 10.64: IPNO, KNOWLEDGE, Needleman-Wunsch, Bottom up only

	TP	FP	Prec	Rec	F-2
M	0.001	0.006	0.1	0.029	0.045
N	0.934	0.021	0.978	0.975	0.976
U	0.005	0.032	0.135	0.28	0.182
Avg	0.895	0.021	0.941	0.94	0.94

Table 10.65: KNOWLEDGE, Needleman-Wunsch, Diachronic only Top-down

	TP	FP	Prec	Rec	F-2
M	0.001	0.007	0.091	0.029	0.044
N	0.934	0.021	0.978	0.975	0.976
U	0.005	0.032	0.135	0.28	0.182
Avg	0.894	0.021	0.941	0.939	0.939

Table 10.66: KNOWLEDGE, Needleman-Wunsch, Combinatorial only Top-down

	TP	FP	Prec	Rec	F-2
M	0.017	0.039	0.304	0.706	0.425
N	0.231	0.004	0.982	0.241	0.387
U	0.009	0.699	0.013	0.52	0.026
Avg	0.222	0.018	0.948	0.257	0.381

Table 10.67: IPNO, KNOWLEDGE, Needleman-Wunsch, Mixing with Combinatorial Top down

Table 10.68: Experiments for mixed rules evaluation on Manually Annotated dataset for Organization

- Knowledge-driven comparison method, with Needleman-Wunsch similarity metric, applying inconsistency removal normalization (NPIR), threshold normalization conservative for positive matching rules and negative matching rules (TNCC), integrating only negative matching top-down rules (IPTO) (table 10.62). The ρ-accuracy is 0.959. If we consider plain rules integration, the ρ-accuracy is the same, but the precision of positive match decreases (table 10.63). This seems to suggest that positive matching rules caused some false positive matching. This aspect will be analyzed more in depth in section 10.2.4.

Comparing the results of matching performances relying on rules relying only on bottom up approach (table 10.64), we can see how the mixed approach including negative matching rules supported a higher recall of negative matching cases. If we consider only top-down diachronic rules, the performances of matching are surprisingly good (table 10.65). The precision and recall of negative matching cell is very higher than

Class	TP	FP	Precision	Recall	F-Measure	ρ-accuracy
Mixed-D	0.026	0.0	1.0	0.026	0.05	0.046
Bottom-up	0.025	0.0	1.0	0.026	0.05	0.046
Top-D	0.005	0.598	0.008	1.0	0.016	0.006
Top-C	0.00	0.0	0.0	0.00	0.00	0.00
Mixed-C	0.027	0.0	1.0	0.027	0.052	0.046

Table 10.69: NYT Dataset, comparing approaches for Organization

the mixed rules approach, probably due to the fact that conservative threshold of negative samples set to 0.5 for diachronic attributes is more relaxed than the one defined in bottom-up rules extraction with conservative threshold normalization. However, top-down diachronic rules perform very greedy positive match classification, resulting in a very low precision compared with mixed rules approach. In fact, the f-measure of positive match is 10 times worst that the obtained with mixed rules and therefore also in terms of ρ-accuracy the performances are little worst (0.95). Considering top-down rules built by combining functional attributes defined in the identification ontology (table 10.66), the positive trend is confirmed, but the overall matching performances are still little worst that the one obtained with mixed rules, if we consider ρ-accuracy. The main defect of matching with these rules is related to the greedy positive match decisions that cause a relatively high number of false positive match. If we consider mixing combinatorial top down rules with bottom up learned rules (table 10.67), the matching performances decrease in terms of negative matching recall. This is due to the conservative approach in the merging processes, decreasing the effectiveness in taking negative matching decisions.

Performing experiments with the NYT dataset for organization, the performances are particularly disappointing. Several configuration performed in the same (bad) way in terms of accuracy. For a matter of consistency, we choose also this time to represent the results obtained comparing entities using the knowledge-driven approach. In particular we present details for the configuration:

- Knowledge-driven comparison method, applying inconsistencies removal normalization (NPIR), applying relaxed threshold for positive matching rules and conservative thresholds for negative matching rules, integrating positive and negative top-down diachronic rules (IPP) and relying on Jaro-Winkler string similarity metric.

The matching performances of all the cases considered is very bad, as show in table 10.69. A deeper analysis on the matched description is required to provide a justification to such bad results. A poor training set could justify performances of

Class	TP	FP	Precision	Recall	F-Measure	ρ-accuracy
Mixed-D	0.876	0.0	1.0	0.876	0.934	0.934
Bottom-up	0.618	0.0	1.0	0.618	0.764	0.759
Top-D	0.854	0.0	1.0	0.854	0.921	0.863
Top-C	0.854	0.0	1.0	0.854	0.921	0.863
Mixed-C	0.618	0.0	1.0	0.618	0.764	0.719

Table 10.70: OAEI 2010 Dataset, comparing approaches for Organization

bottom up and mixed matching, but the fact that also top-down only matching failed, seems to suggest that perhaps there is some problem in the data. Namely, what it is told to be the same, but it is not really the same.

If we consider the OAEI 2010 restaurant dataset, the configuration that performed better is the following:

- Knowledge-driven comparison method, applying inconsistency normalization (NPIC), relying on conservative threshold normalization for both positive and negative matching rules, integrating both positive and negative top-down diachronic matching rules, and relying on Taglink similarity metric. The score in terms of ρ-accuracy is 0.934, as presented in table 10.70.

Comparing the results of different approaches in the definition of rules (table 10.70), we can notice how the application of a mixed rules involving top-down diachronic rules and bottom-up learned rules is the one performing better. All the methods matched with perfect precision, but the recall of mixed rules involving top-down diachronic granted a higher recall. This is probably due to the fact that evaluation set presented inverse-functional properties matching, which often were sufficient to take positive matching decision. The effect of inverse-functional properties is reduced considering top-down combinatorial rules as positive matching rules require the satisfaction of a larger number of attributes (minimum 4, as described in section 8.1).

Method	Person	Organization	Location
Greedy	0.500	**0.384**	0.336
Knowledge	**0.508**	0.379	0.367
Char-RC	0.502	0.384	**0.377**

Table 10.71: Comparison Method Average ρ-accuracy

Method	Person	Organization	Location
Equal	0.443	0.178	0.146
Levenshtein	0.523	0.456	0.406
Euclidean	0.492	0.486	0.364
Jaccard	0.523	0.343	0.320
Jaro-Winkler	0.541	0.494	0.387
Monge-Elkan	0.500	0.246	**0.490**
Overlap	0.483	0.242	0.304
QGram	0.541	0.561	0.398
Smith-Waterman	**0.547**	0.249	0.404
Needlman-Wunsch	0.539	0.317	0.429
Taglink	0.529	**0.614**	0.412

Table 10.72: String Similarity Average ρ-accuracy

10.2.4 Experiments Results Analysis

In the previous sections we presented the results of an extensive, but not complete set of experiments. In particular, through these experiment we aimed at evaluating the feasibility of the proposed approach as a reliable solution to real world entity matching problems. To perform such evaluation, we choose three evaluation sets presenting different characteristics (see section 10.1). Analyzing the results it is possible to assert that both string similarity and comparison method strongly affects both learning process and matching results. At the same time, relying on the best cases analyzed, no best method clearly emerged, nor a string similarity metric showed to be performing particularly better than others. However, in the previous section, we choose to analyze the best cases, without considering overall performances on all the datasets. Therefore, we decided to perform this analysis, and compute the average ρ-accuracy of all comparison methods relying on each of the single similarity metrics, and average ρ-accuracy for each of the similarity metric based on each of the comparison methods. The best combination on the ranked lists is the one selected to work better, in average. In table 10.71 we show that for person, the Knowledge-driven comparison method worked in average slightly better than the other. For Organization, the Greedy and the Char-RC performed equally, whereas for location Char-RC performed better. In table 10.72 we show that for entity type person, the string similarity metric that in average performed better is Smith-Waterman. Considering the entity type Organization, the similarity metric that performed better in average is Taglink, whereas the for entity type location Monge-Elkan.

Comparing the results of the single approaches for rules definition presented in section 10.2.1 and 10.2.2, with the result of the mixing these rules and normalizing them, we can confirm that rules obtained mixing bottom up and top-down supported more accurate matching decision than each of the single approaches. Only considering person OAEI 2010 dataset for entity type person, the experiment executed mixing rules result of combination of attributes defined in the identification ontology performed better than the configuration involving mixed rules learned bottom and defined top-down relying on functional and inverse-functional diachronic properties. Considering the OAEI 2010 datasets, the proposed method performed in an excellent way. The OAEI 2010 dataset for person, the best configuration performed with a ρ-accuracy of 0.988. The F-measure of 9.88 shows that the matching samples that could not be matched were classified as don't know. This score is in line with RiMOM [142] (0.97) the best tool that performed experiment on that dataset at OAEI 2010 initiative[5]. Considering the restaurant dataset, our method produced a score of 0.934 on restaurant dataset. This score is more than 10 points above the results of RiMOM [142] (0.81). This shows that the method can perform well when data perturbation is related to syntactic differences that can be handled by the different string similarity metrics.

name: Edmund Andrews; **occupation**: reporter; **description**: Edmund L. Andrews is a former economics reporter;
isPrimaryTopicOf: Edmund Andrews; **wasDerivedFrom**: Edmund Andrews?oldid=481764669; **wikiPageDisambiguates**: Edmund Andrews (reporter); **wikiPageDisambiguates**: Edmund Andrews (surgeon);

Table 10.73: Examples of Ambiguous descriptions Filtered

Considering the New York Time dataset, the proposed method performed quite well for person, with a ρ-accuracy of 0.879 mixing rules with Top-down diachronic rules, and 0.916 mixing bottom up rules with combinatorial rules. This is the only case where mixing combinatorial rules with bottom up rules provided advantages. The NYT dataset for person is particularly challenging. First of all, the manually maintained link between DBPedia and Freebase are not always updated. In fact, gathering descriptions from the sources lead us to necessity of filtering ambiguous links. In fact, several *owl:sameAs* when resolved lead to a 'disambiguate' description, or an empty description on DBPedia side (see table for example 10.73). In many cases, we had to refresh descriptions following *redirect* links. Furthermore, curiously we several *owl:sameAs* linked descriptions of entities declaring types mapped to different classes in the identification ontology, or not declaring any specific type. At this point, we do not deal with these descriptions, and simply filter them out from the evaluation set.

[5] `http://www.instancematching.org/oaei/imei2010/pr.html`

In all, we discard about 500 description corresponding to the 11.6% of the samples. It is important to notice that the non perfect recall that lead to an f-measure of 0.879 is due to structural heterogeneity affecting the considered descriptions. Consider the descriptions presented in table 10.74 and 10.75. The descriptions are clearly matching, but the details about the data of birth and date of death do not allow to take positive matching decision. Therefore, a *don't know* decision is taken. In some cases, the rules supported a negative matching decision despite, which in this case has to be considered a mistake. Consider for example the descriptions in table 10.76 and 10.77. Our solution decided it is not a match, but the error is due to the error in data. In fact the birthdate is different, and only one of the two can be right. Knowledge-based solution is not robust with respect to this type of error. Furthermore, the name attributes are particularly different, thus it is acceptable as error. All the false negative match analyzed in the dataset are due to errors in the data. Therefore, we can assume that the method works well if data are correct.

name: Wolfgang Wagner; **last_name**: Wagner; **birthdate**: 1919; **description**: Wolfgang Wagner (30 August 1919 21 March 2010); **date_of_death**: 2010; **first_name**: Wolfgang **domain_tag** : Category:People from Bayreuth **domain_tag** : Category:Wagner family **domain_tag** : Category:German people of English descent **domain_tag** : Category:German opera directors **website** : `http://www.imdb.com/title/tt0111475/`

Table 10.74: http://dbpedia.org/resource/Wolfgang_Wagner

name: Wolfgang Wagner; **last_name**: Wagner; **birthdate**: 1919-08-30; **description**: Wolfgang Wagner (30 August 1919 21 March 2010); **date_of_death**: 2010-03-21; **first_name**: Wolfgang **occupation** : Theatre Director **birthplace** : Bayreuth **domain_tag** : Opera Director **domain_tag** : Category:German opera directors **gender** : Male

Table 10.75: http://www.freebase.com/view/en/wolfgang_wagner

name: Ali Khamenei; **last_name**: Khamenei; **birthdate**: 1939-07-17; **description**: Ayatollah Seyed Ali Hosseini Khamenei ...; **first_name**: Wolfgang **birthplace**: Ali; **occupation**: Politician; **city_of_residence**: Mashhad **occupation**: President - President of Iran; **affiliation**: Islamic Republican party; **domain_tag**: Politician; **birthyear**: 1939; **birthmonth**: 6; **day_of_birth**: 17; ...

Table 10.76: http://www.freebase.com/view/en/ali_khamenei

Analyzing the organization dataset, we can see that performances are particularly disappointing. The explanation to such poor results is in the difference between the nature of the training set, and the NYT evaluation dataset. The training set we evaluated produce the following entity matching rules:

```
name < 0.96912
```

```
     and city < 0.758313
     and street_address < 0.846489 then NON_MATCHING

name >0.992982
     and city > 0.758313
     street_address > 0.846489 then MATCHING

email_address_hashcode >= 1.0 then MATCHING
public_institutional_id >= 1.0 then MATCHING
picture_URL >= 1.0 then MATCHING
website >= 1.0 then MATCHING
phone_nr >= 1.0 then MATCHING
fax_nr >= 1.0 then MATCHING
email_address >= 1.0 then MATCHING
```

These rules are very simple, maybe even too simple due to the limited size of the training set. However, they seem to be reasonable, and not too strict. However, looking at the samples of data, none of the compared description presented together the attributes necessary to take a matching decision. Consider for example the descriptions presented in table 10.79 and 10.79. Therefore, looking at the descriptions, it is possible to see how the rules considered cannot be applied, as the description, besides the name, have very few attributes that can be matched considering their semantics. This is a limit of our knowledge based solution, given the fact that in this context matching only considering the name would be lead a high recall and precision. Nevertheless, we believe that it would do it for the wrong reason, and not in a reliable way. Therefore, we stick with our poor performances on this dataset, but hopefully better performances on other types of datasets. All the entities manually analyzed present the structure and the types of attributes defined in the tables described. If we consider the evaluation with the NYT dataset for location, also the performances of the best case are below the performances declared for examples in [118]. Analyzing our best result thou, we realized that the best classification obtained relying on greedy comparison weighted with Charachter-based Relative Completeness relied on very extremely simple rules:

name: Seyed Ali Hosseini Khamenei; **name**: Seyyed Ali Hosseini Khamenei; **last_name**: Khamenei; **birthdate**: 15-07-1939; **description**: Grand Ayatollah Sayyed Ali Hosseini Khamenei ...; **first_name**: Seyyed Ali Hosseini; **occupation**: 3rd President of Iran; **birthplace**: Mashhad, Iran; **member_of**: Usuli; **domain_tag**: Category:Presidents of Iran; **gender**: Male; **birthyear**: 1939; **birthmonth**: 6; **day_of_birth**: 15; ...

Table 10.77: http://dbpedia.org/resource/Ali_Khamenei

name: 1st Constitution Bancorp; **controls**: 1st Constitution Capital Trust II; **has_members**: Charles S Crow III; **has_key_people**: Charles S Crow III; **description**: 1st Constitution Bancorp (NASDAQ:FCCY) is the New Jersey holding company ... ; **organization_type**: Public company; **occupation**: 3rd President of Iran; **street_address**: 1285 WESTHAVEN CIRCLE, Vail, Colorado, United States of America 81657; **organization_type**: Public Company; **street_address** : Cranbury CDP, New Jersey **street_address** : 2650 ROUTE 130, Cranbury, New Jersey, United States of America 08512 **activity_sector** : Financial Services **foundation_date** : 1989 ...

Table 10.78: http://www.freebase.com/view/en/1st_constitution_bancorp

```
location_name =< 0.714286 NON_MATCHING
```

name: 1st Constitution Bancorp; **activity_start_year**: 1989; **activity_sector**: Bank; **description**: 1st Constitution Bancorp is the New Jersey holdin ...; **has_key_people**: Robert F. Mangano; **has_location**: Cranbury, New Jersey; **organization_type**: Public Company; **domain_tag** : Category:Banks established in 1989 **domain_tag** : Category:Companies based in Middlesex County, New **domain_tag** : Category:Companies listed on NASDAQ **domain_tag** : Category:Banks based in New Jersey **slogan** : Community Banking With You In Mind ...

Table 10.79: http://dbpedia.org/resource/1st_Constitution_Bancorp

location_name: Mont Saint-Michel de Bras-Part; **location_name**: Montagne Saint-Michel; **location_name**: Mont Saint-Michel-d'Arre; **first_level_administrative_parent**: Bretagne; **forth_level_administrative_parent**: Saint-Rivoal; **second_level_administrative_parent**: Finistre; **third_level_administrative_parent**: Arrondissement de Chteaulin; **latitude**: 48.35; **country**: France; **longitude**: -3.95; **geocoordinate**: 48.35 -3.95 **picture_URL**: `http://www.geonames.org/2978007/mont-de-saint-michel.html` ...

Table 10.80: http://sws.geonames.org/2978007/

```
location_name >= 0.888889 MATCHING
picture_URL >= 1.0 MATCHING
website >= 1.0 MATCHING
coordinate_geometry >= 1.0 MATCHING
geocoordinate >= 1.0 MATCHING
```

The don't know classification is the due to the cases where names are in between the range of 0.71 and 0.88. We do not believe that this approach to classification, despite the result, is suitable for a reliable entity matching solution. In fact, considering the manual annotated dataset, this type of classification produced very poor classification results due to several homonym locations and thus false positive match decisions. If we perform entity matching experiment relying on the configuration that performed better on the manual dataset of locations, we obtain a way larger number of "don't know" classified samples, due to these more restrictive rules:

location_name: Mont Saint-Michel; **location_name**: Montagne Saint-Michel; **location_name**: Mont Saint-Michel-d'Arre; **is_contained_by**: Manche; **location_type**: Governmental Jurisdiction ; **location_type**: Tourist attraction; **location_type**: Island; **latitude**: 48.6356; **timezone**: Central European Time; **longitude**: -1.5111; **geocoordinate** : 48.6356, -1.5111 **picture_URL** : `http://api.freebase.com/api/trans/image_thumb/en/` ...

Table 10.81: http://www.freebase.com/view/en/mont_saint-michel

```
latitude =<0.733333
    and location_name =< 0.940117 NON_MATCHING
longitude >0.849817
    and location_name>0.940117
    and latitude > 0.913333 MATCHING
longitude =<0.849817 NON_MATCHING
picture_URL >=1.0 MATCHING
website>=1.0 MATCHING
coordinate_geometry >= 1.0] MATCHING
geocoordinate >= 1.0 MATCHING
```

The large number of "don't know" classified samples is due syntactical difference in the representation of the latitude and longitude attributes. In fact, Geonames, DBpedia and Freebase tend to represent these important attributes in a very heterogeneous way, making it hard to perform matching relying on string similarity metrics. Consider for example the samples in tables 10.80 and 10.81. As shown in the description, both names and coordinate are quite different both in terms of granularity and also in terms of latitude. This is probably due to the fact that geo-coordinates are defined according to different coordinate systems and thus produce very different results although referring to the same point. Honestly, if it was not because of the famous location as a touristic attraction, we would not dare saying these two descriptions refer to the same location, even thou it is clearly possible. Considering the descriptions in tables 10.81 and 10.82 the comparing seems easier, even thou in this case the location names are quite different from a syntactic perspective.

location_name: Le Mont Saint-Michel; **is_contained_by**: Basse-Normandie; **is_contained_by**: Lower Normandy; **description**: Mont Saint-Michel is a rocky tidal island and a ...; **location_type**: Communes of Manche; **latitude**: 49; **latitude**: 48.636; **coordinate_geometry**: POINT(-1.5114 48.636) ; **longitude**: -2; **longitude**: -1.5114; **geocoordinate** : 48.636 -1.5114 **postal_code** : 50116 **country**: France ...

Table 10.82: http://dbpedia.org/resource/Mont_Saint-Michel

10.3 Comparing with to FBEM Matcher

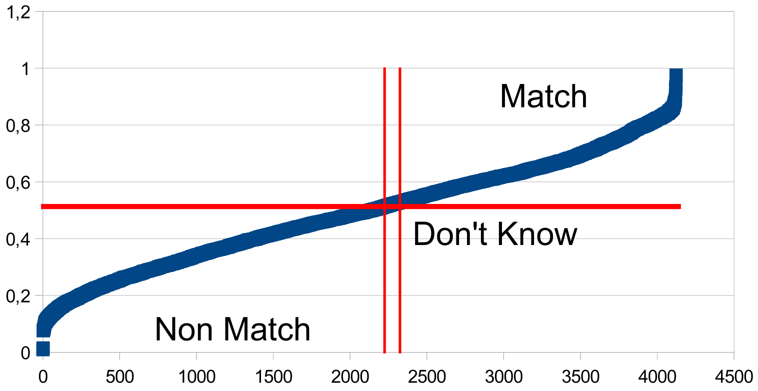


Figure 10.1: Distribution of FBEM scores NYT

In this section we present a comparative study with the feature based entity matching solution (FBEM) described in [139]. The FBEM solution is the default matching module of the Okkam Entity Name System, and is aimed at solving the problem of entity matching under the conditions we considered in this context, that are semantic and structural heterogeneity to match descriptions on the web. The FBEM solution is a distance-based solution, combining ontological knowledge with probabilistic methods. The outcome of the FBEM algorithm is not a matching decision then, but rather a score of similarity that can be used to rank objects. In order to compare the results of the FBEM solution in a fair way, we decided use the training set to learn the similarity thresholds that maximizes the F-measure of the classification of the training set. In order to do so, we implemented a simple algorithm that decreases the matching threshold looking for the f-measure classification for positive matching classification. Complementary, iteratively we learned the similarity threshold for negative matching cases by incrementing the threshold with an upper bound defined by the positive matching threshold previously defined. We repeated the iterative threshold learning process for 100 times, incrementing and decrementing the thresholds of 0.01 at each iteration. We are aware that this method is very basic, but it serves the purpose of defining a comparison between Fingerprint match and the proposed solution. Notice that we applied the same process to the solution we propose.

For a matter of space we present only the results on the evaluation of the NYT datasets that are the one that challenged the most our approach. In table 10.83 the results of the FBEM algorithm classification of the New York Times dataset for person. The iterative threshold algorithm supported the definition of similarity thresholds of 0.53 for positive matching decisions, and 0.52 for negative matching decisions. As shown in table 10.83 the FBEM algorithm classified with a good precision, but the recall is quite low despite the positive matching similarity threshold is relatively low. An explanation to this performance can be given analyzing the graph presented in figure 10.1. Evaluating fingerprint, we performed a slightly different experiment by maximizing don't know score satisfaction. In fact, positive and negative match score are always either 0.0 or 1.0. Therefore, maximizing the satisfaction of don't know F-measure, we defined a threshold of 0.52 which is very similar to the one defined for

class	TP	FP	Precision	Recall	F-measure
match	0,432	0,000	1,000	0,433	0,605
non-match	0.002	0,548	0,004	1,000	0,007
don't know	0.000	0,018	NaN	NaN	NaN
weighted avg	0,432	0,001	0,998	NaN	NaN

Table 10.83: FBEM Threshold Based comparison NYT

FBEM. With this threshold, the experiment produced the results presented in table 10.84. The fact that the solution we proposed defines a clear, boolean, clause for taking matching decision allows to improve recall relying on threshold management, keeping the precision in negative classification. In fact, with respect to FBEM, the solution we propose defined a much lower amount of false negative matching decision which are due to errors in the data. This aspect is further highlighted analyzing figure 10.2.

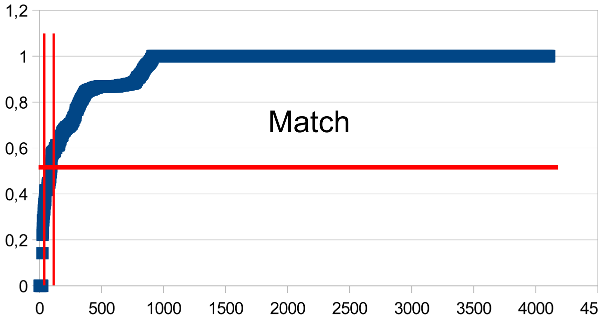


Figure 10.2: Fingerprint Score Distribution

Therefore, with this experiment, we proved that for the entity type Person, the fingerprint method provides a robust solution performing better than FBEM in all aspects of the comparison. In fact, Fingerprint Match, selecting the attributes for string comparison based on the rules, reduces considerably the number of expensive string matching operations. This allows to take matching decision with average time of 7.8 ms per comparison on a regular laptop, versus the 406 ms in average required by FBEM on the same laptop.

class	TP	FP	Precision	Recall	F-measure
match	0,973	0,000	1,000	0,975	0,987
non-match	0,000	0,009	0,000	0,000	NaN
don’t know	0,000	0,017	0,000	NaN	NaN
weighted avg	0,971	0,001	0,998	NaN	NaN

Table 10.84: Fingerprint Threshold Based comparison NYT for Person

Part IV

Conclusions and Appendix

Chapter 11

Conclusions and Future Work

In this work, we defined, implemented and evaluated a knowledge-based framework for the solution of the entity matching problem in the open and wide context of the (Semantic) Web.

As a first step, we defined and formally validated a lightweight ontology, defining 3 main entity types: person, location and organization. For these classes, we defined respectively 39, 31 and 43 features. We formally grounded the definition of contextual mappings used to harmonize the semantic of existing vocabularies and schema with the defined ontology. The person entity type was mapped with 23 equivalent classes defined in other ontologies, and 207 other types that could be considered subclasses. The properties of entity type person were mapped to 999 attributes defined on other ontologies and schemas. The organization entity type was mapped with 20 equivalent classes, and 2551 subclasses (a large part of these from Yago ontology). The features of the organization type were mapped with 1125 attributes defined in different schemas and ontologies. The location entity type was mapped with 22 equivalent classes, and 2325 subclasses. The features of entity type location were mapped with 368 properties defined in other ontologies. Each of the considered features was analyzed using formal ontology tools to annotate part of them with meta-properties supporting the definition of entity matching rules. In particular, we analyzed the defined features annotating them as functional and inverse-functional diachronic properties, establishing whether these could be suitable as diachronic identity criteria for the considered entity types.

As a second step, we provided a formal definition of entity matching rules using the Three Value Logic of Kleene and some principles of Intuitionistic Logic. This was done with the goal of defining rules to compose an equational theory which would encompass the Open World Assumption. In fact, the solution of the entity matching problem in the open and wide context of the Web impels us to consider also the case where no reliable entity matching decision can be taken, and thus a third *unknown* case must be considered. Together with the formalization of the rules, we defined some tools that would allow us to combine these rules in different ways. In particular, we defined a pragmatic principle of rules subsumption that was used to guide the process of the merging of rules. We also defined several principles of normalization of the rules, supporting the definition of conservative, reliable matching rules.

As a third step, we designed and implemented a laboratory experiment to learn entity matching rules. This experiment relied on the Decision Tree classifier, a regression technique to extract entity matching rules given a training set of labeled samples. In this experiment, people of different age, gender and education were asked to undertake a set of simple entity matching tasks on pairs of descriptions. The objective of this experiment was to collect feedback about matching decisions. This allowed us to create a training set to support the elicitation of the *matching knowledge* people employed in taking matching decisions. The descriptions used in the experiment were collected from heterogeneous sources using randomly selected names. The creation of single comparison tasks relied on a simple blocking system based on a Apache Lucene and *tf/idf*. When presented to the user, descriptions collected were shuffled in the order of appearance of the attributes, to force a complete scan of the descriptions, and reduce the effect of application of cognitive heuristic related for evaluation of few attributes only. The training set built in this way presented 7405 labeled sample pairs, involving 2094 different descriptions for entity type person. The training set for location type presented 2310 labeled sample pairs, involving 1013 different descriptions. Considering entity type organization, 5064 sample pairs were labeled, involving 3051 descriptions. This datasets were used to elicit entity matching rules to support entity matching decision.

The last part of the work described the implementation and evaluation of a software program named Fingerprint Match relying on a combination of rules that resulted from two complementary processes. Firstly, we defined entity matching rules relying on ontological analysis about the features contained in the identification ontology. Secondly, we defined entity matching rules as the result of a bottom-up, machine learning supported process. In order to evaluate matching performances coherently with the open world assumption, we integrated the traditional accuracy evaluation metrics (precision, recall and f-measure) with a new metric named ρ-accuracy that weighted differently, but symmetrically, false positive matching and false *don't know* matching classification. In chapter 10.2.4 we presented the results of experiment considering the following objectives:

- compare the results related to the adoption of different any similarity metrics considered;

- compare the results related to 3 different methods of comparison (Greedy approach, Simple Knowledge-driven approach and Greedy with Relative Completeness Character-based;
- compare the results 2 different rule inconsistency normalization approaches (inconsistency removal and inconsistency normalization)
- compare the results of all possible threshold normalization processes;
- compare the results of positive only, negative only and plain top-down diachronic rules integration;
- consider the impact of binary and multiclass classification method;

The combination of these factors were tested on all the evaluation datasets, for the three entity types considered. At the end of the evaluation process, considering experiments on bottom-up extracted rules, top-down defined rules and their combinations, we can count the execution of more than 38000 experiments. In analyzing the results of the experiments, the following lessons were learned:

- The first lesson is that *one size does not fit all.* This may be obvious, but it is clear that defining a single solution that works with every dataset is not feasible. However, the framework defined does not depend specifically on any factor, such as similarity metric, comparison method, ontologies, mappings or rules. In fact, different combinations of these tools can be applied to define different configurations, as shown in the chapter 10.2. Nevertheless, the experiment we executed shows that in terms of ρ-accuracy, Smith-Waterman similarity metrics in average works slightly better than the others for matching descriptions of persons. Regarding organization, Taglink is the similarity metric that in average best performed, whereas Monge-Elkan was the best for locations.
- The integration of top down rules formed by diachronic functional and inverse-functional properties with the bottom-up learned rules showed to be more effective than the application of each single set of rules separately. However, it is important to notice that best matching performances were achieved integrating only positive matching top-down rules. In fact, negative rules based on functional diachronic properties showed to be too restrictive due to frequent errors in the evaluation datasets. At the same time, rules constructed combining functional attributes defined in the ontology showed to be precise in negative matching decisions, as these

were less restrictive. Therefore, we have to test the definition of less restrictive top-down entity negative matching rules.

- Another interesting lesson is that a knowledge-driven approach to entity matching can provide benefits if compared to simple greedy comparison. In fact, by exploiting the meta-properties used to annotate the features in the ontology, we could apply different matching techniques to each of them. This supported more precise learning and matching resolutions.

- Among the configurations we evaluated, when considering entity type person the most effective rule normalization process was the one that applied inconsistency removal. In particular when combined with a conservative thresholds normalization. Whereas, for the other types, the normalization process that proved to be more effective was inconsistency normalization. This seems to suggest a correlation between the numbers of samples in the dataset, and necessity of defining more or less conservative rules. Namely, the smaller training sets required conservative normalization of the rules to support reliable matching decision, whereas the larger training set demanded some sort of pruning of the rules. To confirm this correlation, future experiments will have to evaluate the adoptions of filters, or training set partitions and combinations.

- Relying on multi-class classification seems to penalize the rules learning process compared to simple binary classification. Intuitively, the *don't know* labeled samples should support the learning of more conservative, precise rules. However, the subtle nature of the *don't know* labeled samples may have negatively affected the learning of positive matching rules, as these were *hidden* among the don't know cases. We will further investigate this issue to better understand how to leverage the *don't know* labeled samples.

- Performing entity matching on entity type location requires the implementation of special purpose techniques to match geo-coordinates. In fact, syntactic string matching proved not to be effective in matching these types of attributes. The possibility of specializing matching solution for single attributes type is an important feature of the knowledge-based solution we propose.

- Performing entity matching on entity type organization also requires the implementation of special solution for matching addresses. This problem is known to be complicated, but incremental specialization of matching on street address can lead to improvements in matching performances.

- When compared with the Feature Based Entity Matching algorithm currently employed as default matching module in the Okkam Entity Name System[1], the Fingerprint Match solution was proven to be more effective and efficient, showing its reliability also in terms of producing a score. Given this experiment, it is our intention to deploy the Fingerprint Match solution as matching module for the ENS.

The experience accumulated in conceiving and implementing the solution proposed in this work allows us to look positively at the future, as the solution was proved to be both feasible and effective. However, there are still some issues that have to be unfolded in order incrementally improve the quality of the implementation. For this reason, we aim to continue investigating for more innovative solutions:

- the manual definition of mappings for semantic harmonization is cumbersome and potentially error prone. In fact, in the dataset we often found attributes whose value was not semantically correct (e.g. birth-date: Trento, Italy). Other times, we found errors related to overloading in the interpretation of the semantic of attributes, as for example *begin* as the date of birth of a person. All these issues make also the definition of manual mappings for semantic harmonization error prone. Therefore, in the near future we plan to define an hybrid system for automatic guessing of attributes type given a value. The system will have to rely on a combination of statistical and syntactical methods, to compensate the weakness of each of the approaches.

- the main goal of an ontology is to be a shared representation of a domain. So far, we conceived the ontology as a parameter of the framework, without considering the need of sharing it. Therefore, we plan to design and deploy a wiki-like web site that would allow subscribers in proposing and discussing about possible evolutions of the ontology. The idea is to make the web site a point of reference for the collection and definition of features and mappings of existing ontologies to the defined features for the solution of the entity matching problem on the web. The wide set of mappings defined is sufficient to classify the ontology a well linked vocabulary. The next step is to make it openly available.

- The framework we defined in this context is suitable to solve the problem of entity matching when a sufficient amount of information is considered. Therefore, to exploit the proposed solution at its best, we'll develop a plug-in for the Open Refine tool to support the reconciliation of datasets with the Entity Name System.

[1] It is possible to access the APIs at `http://api.okkam.org`

We believe that implementing and sharing this plug-in extension for Open Refine would also foster the collection of precious feedback to support the bottom-up rules extraction process.

We believe that the main objective achieved with this work is to prove how a knowledge-based solution can be suitable to solve the problem of entity matching in the context of the Web. We are convinced that the defined framework will allow to incrementally define more effective and efficient techniques, specializing matching of specific attributes on one side, and investigating possible further meta-properties of the features defined in the identification ontology on the other. Despite more experimental evaluation is necessary, we believe that the experimental evaluation presented in this work allow us state that the successfully reached our objectives. The lessons learned will be the base of future improvements. Furthermore, the integration of the implementation of this solution as an effective module for entity matching in the Entity Name System will support the definition of a more precise and efficient reconciliation service, enabling in principle the *okkamization* of diverse and heterogeneous data sources, moving a step further in the realization of the Web of Entities described in [28], and possibly in the definition of a better Semantic Web.

Chapter 12

Acknowledgments

The accomplishment of this work is the result of an effort that would not have been possible without the collaboration and support of many great people that are part of my life. All of them participated somehow to make it possible in many different ways.

First of all, I want to thank professor Paolo Bouquet for being my advisor for so many years. Every word, every discussion, and even every moment of silent trust contributed significantly not only in the realization of this work, but most importantly pushed me ahead in a path of personal growth and maturation. I think I still have a lot to learn from him to become an accomplished researcher, but I believe the lessons learned will allow me to get there one day. Besides Paolo, I have to thank all the brilliant researchers and collaborators that were my colleagues in the past years. Among the most influential, I have surely to thank doctors Barbara Bazzanella for contributing with fruitful discussions to the realization of this work; Heiko Stoermer for helping me in growing in terms of self-discipline and organization; Massimiliano Vignolo for the philosophical discussions about the problem of identity and reference and the Web; and George Giannakopoulos for introducing me to the world of machine learning. I want to thank also professors Craig Knoblock and Pedro Szekely for welcoming and advising me while visiting the Information Sciences Institute of the University of Southern California. My visit at ISI was not very long, but it was essential to find solutions to many practical issues related to the application of machine learning techniques.

I want also to thank my future wife Paula, for being such a great companion along this journey. Thanks for patiently listening my concerns, supporting me in the difficult moments, and sharing the joyful ones. Thanks for taking me to take a GROM ice-cream even late at night when I needed some fresh air after many hours in front of the computer. You were always there for me, I will always be there for you. Life can be much more complicated than writing a PhD thesis, but it will be a pleasure to live it

with you. There will be further more challenges we'll have to deal with. The important thing is that we do it together.

I have to thank also my family and friends, for supporting me in these years, and for reminding me that there is life beyond the PhD program. I want to thank my parents for constantly helping me in these long years of studies, accepting decisions even when these led me to live abroad. You made me what I am, and even knowing I can improve in many aspects, I am very proud of what I have become, and I have to thank you for this. I have to thank all my friends, in particular Jacopo and Simone for being such great buddies and *consiglieri* in this phase of my life. We had many discussion about sport, music, women, and shared improvised dinners, beers and sips of whiskey. I truly hope this will never change, as I will always need good friends as your are.

Last but not least, I want to thank Flavio, Nicola, Fabio, Lorenzo, Daniele, Valentina and Pavlo at the Okkam Labs for being such great colleagues, sharing long working hours in office, and helping me specially in the realization of the experiments for this thesis. We make a great team, and I am sure there is a great future in front of us!

Appendix A

Appendix A: Semantic Harmonization Mappings

http://www.freebase.com/schema/location/location
http://schema.org/Place
http://dbpedia.org/ontology/Place
http://dbpedia.org/class/yago/YagoGeoEntity
http://www.w3.org/2006/03/wn/wn20/instances/synset-location-noun-1
http://models.okkam.org/identification-ontology.owl#location
http://umbel.org/umbel/rc/Place
http://www.geonames.org/ontology#Feature
http://www.opengis.net/gml/_Feature
http://sw-portal.deri.org/ontologies/swportal#Location
http://data.archiveshub.ac.uk/def/location
http://purl.org/dc/terms/Location
http://purl.org/goodrelations/v1#Location
http://www.ontotext.com/proton/protontop#Location
http://purl.org/vocab/frbr/core#Place
http://rdvocab.info/uri/schema/FRBRentitiesRDA/Place
http://vivoweb.org/ontology/core#GeographicLocation
http://data.press.net/ontology/stuff/Location
http://www.w3.org/2006/vcard/ns#Location
http://semanticweb.cs.vu.nl/2009/11/sem/Place
http://www.freebase.com/schema/sports/sports_facility
http://www.freebase.com/schema/architecture/structure
http://www.freebase.com/schema/architecture/venue
http://www.freebase.com/schema/architecture/skyscraper
http://www.freebase.com/schema/architecture/building
http://dbpedia.org/ontology/ArchitecturalStructure
http://purl.org/acco/ns#Suite
http://www.freebase.com/schema/location/country
http://schema.org/Museum
http://vivoweb.org/ontology/core#GeographicRegion
http://dbpedia.org/ontology/Museum
http://purl.org/acco/ns#Hotel
http://purl.org/acco/ns#Resort
http://dbpedia.org/ontology/Building
http://dbpedia.org/class/yago/ArtDecoBuildingsInCalifornia
http://purl.org/acco/ns#House
http://umbel.org/umbel/rc/Location_Underspecified
http://umbel.org/umbel/rc/PopulatedPlace
http://dbpedia.org/class/yago/BridgesInNewMexico
http://dbpedia.org/class/yago/BridgesCompletedIn1965
http://dbpedia.org/class/yago/GeoclassBridge
http://dbpedia.org/class/yago/IslandsOfThePacificOcean
http://dbpedia.org/class/yago/ValleysOfWales
http://dbpedia.org/class/yago/GeoclassAirfield
http://umbel.org/umbel/rc/Island
...

Table A.1: Entity Type Mappings for Location

http://schema.org/Person
http://dbpedia.org/ontology/Person
http://xmlns.com/foaf/0.1/Person
http://www.freebase.com/schema/people/person
http://www.freebase.com/schema/en/human
http://umbel.org/umbel/rc/Person
http://www.w3.org/2006/03/wn/wn20/instances/synset-person-noun-1
http://d-nb.info/standards/elementset/gnd#Person
http://swrc.ontoware.org/ontology#Person
http://vocab.data.gov/def/drm#Person
http://rdvocab.info/uri/schema/FRBRentitiesRDA/Person
http://www.w3.org/2000/10/swap/pim/contact#Person
http://purl.org/vocab/frbr/core#Person
http://www.ontotext.com/proton/protontop#Person
http://purl.org/ontology/po/Person
http://voag.linkedmodel.org/voag#Person
http://dati.camera.it/ocd/persona
http://models.okkam.org/identification_ontology.owl#person
http://models.okkam.org/identification-ontology.owl#person
http://www.okkam.org/ontology_person1.owl#Person
http://www.okkam.org/ontology_person2.owl#Person
http://umbel.org/umbel/rc/MusicalPerformer
http://dbpedia.org/ontology/Artist
http://dbpedia.org/class/yago/UnitedStatesArmySoldiers
http://dbpedia.org/class/yago/MCARecordsArtists
http://dbpedia.org/class/yago/LivingPeople
http://dbpedia.org/class/yago/PeopleFromFrioCounty,Texas
http://dbpedia.org/ontology/MusicalArtist
http://dbpedia.org/class/yago/PeopleFromSanAntonio,Texas
http://dbpedia.org/class/yago/AmericanCountrySingers
http://dbpedia.org/class/yago/PeopleFromAtascosaCounty,Texas
http://dbpedia.org/class/yago/AmericanMaleSingers
http://www.freebase.com/schema/book/author
http://dbpedia.org/class/yago/AmericanTelevisionActors
http://dbpedia.org/class/yago/Actor109765278
http://www.w3.org/2006/03/wn/wn20/instances/synset-actor-noun-1
http://dbpedia.org/class/yago/AmericanFilmActors
http://dbpedia.org/class/yago/NobelLaureatesWithMultipleNobelAwards
http://dbpedia.org/class/yago/NobelPeacePrizeLaureates
http://dbpedia.org/class/yago/UnitedStatesNavyOfficers
http://dbpedia.org/class/yago/AmericanStageActors
http://dbpedia.org/class/yago/LaSalleUniversityAlumni
http://dbpedia.org/class/yago/ActorsFromPennsylvania
http://dbpedia.org/class/yago/AmericanPeopleOfIrishDescent
http://dbpedia.org/class/yago/SecondCityAlumni
http://dbpedia.org/ontology/Politician...

Table A.2: Entity Type Mappings for Person

http://dbpedia.org/ontology/Organisation
http://schema.org/Organization
http://www.freebase.com/schema/organization/organization
http://umbel.org/umbel/rc/Organization
http://www.w3.org/2006/03/wn/wn20/instances/synset-organization-noun-1
http://models.okkam.org/identification-ontology.owl#organization
http://xmlns.com/foaf/0.1/Group
http://xmlns.com/foaf/0.1/Organization
http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Organization
http://sw-portal.deri.org/ontologies/swportal#Organization
http://www.aktors.org/ontology/portal#Organization
http://www.w3.org/ns/org#Organization
http://purl.org/biotop/biotop.owl#Organization
http://www.ontotext.com/proton/protontop#Organization
http://voag.linkedmodel.org/voag#Organization
http://umbel.org/umbel#Organizations
http://vocab.data.gov/def/fea#OrganizationEntity
http://www.w3.org/2006/vcard/ns#Organization
http://data.press.net/ontology/stuff/Organization
http://dbpedia.org/class/yago/OrganizationsEstablishedIn1935
http://dbpedia.org/class/yago/OrganizationsEstablishedIn1934
http://dbpedia.org/class/yago/CompaniesBasedInMoscow
http://vivoweb.org/ontology/core#Consortium
http://dbpedia.org/class/yago/DefunctCompaniesBasedInPennsylvania
http://dbpedia.org/class/yago/CompaniesBasedInMaconCounty,Illinois
http://dbpedia.org/class/yago/ConsumerOrganizations
http://dbpedia.org/class/yago/CompaniesBasedInBonn
http://vivoweb.org/ontology/core#AcademicDepartment
http://www.freebase.com/schema/government/government
http://dbpedia.org/class/yago/AmericanFootballTeamsInNewYork
http://www.ontotext.com/proton/protonext#PoliticalEntity
http://www.freebase.com/schema/music/musical_group
http://dbpedia.org/class/yago/CompaniesBasedInMemphis,Tennessee
http://dbpedia.org/ontology/Newspaper
http://dbpedia.org/class/yago/BanksOfJapan
http://dbpedia.org/class/yago/BanksEstablishedIn1882
http://www.w3.org/2006/03/wn/wn20/instances/synset-union-noun-1
http://www.freebase.com/schema/metropolitan_transit/transit_system
http://www.ontotext.com/proton/protonext#GeopoliticalOrganization
http://www.ontotext.com/proton/protonext#InternationalOrganization
http://schema.org/MusicGroup
http://dbpedia.org/class/yago/Charities
http://dbpedia.org/class/yago/LGBTOrganizationsInTheUnitedStates
http://dbpedia.org/class/yago/GayMen'sOrganizations

Table A.3: Entity Type Mappings for Organization

Appendix B

Appendix B: Dataset Collection

Table B.2: Queries used to collect samples of Person descriptions: query (nr of samples retrieved)

Parc Ami (7) Konya (53) Bullion Township (13) Tom Smith Lake (17) Sukhar Matak (7) Church Dome (32) Ndougou Department (8) Village of Terrace Park (18) Curtis Peaks (12) Point (237) Tokat Province (22) Hamlet Hill (31) Nansimo Point (3) Chinook Trough (8) Nankai Trough (4) Matak (63) BasseBanio Department (7) Falkland Trough (9) Tom Ka (5) Pennant Trough (12) Mansfield (127) Pliska Ridge (9) Deal Park (11) Zaventem (23) Mount Elephant Lake (21) Pibaore Department (4) Alto Rio Doce (13) Bodega Butte (13) South Bend (47) Oak Ridge Township (19) Pizni (4) Pontian Besar (12) Church Lake (34) Zwevegem (15) Ouerdanine (2) Back Cap (28) Pulau Natuna Besar (3) Kampung Damak (8) Little (225) Rock Springs historical Township (4) Sidr (111) Kampung Dengkil (22) Kyriad Lausanne (8) Orra (18) Kyriad Bergerac (4) Cincinnatus (45) Dixons Corners (3) Kyriad Hotel at the Disneyland Resort Paris (2) Ban Ko (54) Teascu din Deal (2) Laguna el Molina (5) Trough (145) Belmont Township (31) Walachia (10) Golden Glen Township (11) Estate Little Saint Thomas (8) Brialmont Cove (4) Paili (24) Kloulklubed Hamlet (2) Jafr Bak (4) Besteda Bar (20) Patras (72) Queens Court (32) Ban Chak Khok (24) Qobustan (20) Soraocco (2) Cumberland County (18) Oki Trough (3) Peaks (190) Chatswood Oval (8) Kn Kan (25) Matam Department (25) Norfolk Ridge (41) Galesburg Township (29) Tarrant Hinton (12) Grand Parc (22) Qadm (9) High Rock District (11) Belmont (107) Potlogeni Deal (5) Powder River County (22) Kampung (191) Lac TiGris (3)
Kingman Township (10) Portage des Camps (6) Casuarina Point (28) City of Deer Park (15) Tom Green County (17) Al (226) Delaware Department of Transportation Maintenance Area (2) Durango High School District (4) Derang Madng (5) Bodega (166) Flor Airport (19) Prke Shahr (36) Craiova (48) Mutto (8) Tarrant (177) Scioto Ambulance District (3) Playa de la Bodega (4) Consul (39) Kampung Teris (8) La Caleta (12) Sta (84) Kampung Genting (33) Darreh Bak (21) Empire Corners (31) Laguna (219) Techirghiol (21) Oued Agla (2) Commune de Kouinine (3) Coxheath (37) Chikaskia Township (3) Bodega Rock (13) Oued Naam (4) Church Park (24) Church Meadows (27) Nasrallah (23) Kawstah Bn (34) Punta Negra (29) Town of Little River (22) McDonald Court (34) Ninnescah Township (21) Ndolou Department (4) Sagami (43) City of Little Flock (6) Town of Marengo (10) Cap Senino (10) Blank Peaks (8) Desa Sukaharja (9) Lago Gole (15) Church (238) Teiu (47) Gris (183) Jerkuh (26) Hamilton Township (59) River Township (35) Smith Peaks (23) Ban En (11) Comestock Corners (2) Church Mountain (28) Tarrant Gunville (3) Kilis (35) Colonia (103) Four Corners Airport (17) Parc Alexander (40) Uncle Tom Lake (9) Rio Marina (45) Stan (225) Kampung Teluk Ramunia (22) Kampung Jawa (60) Golden Valley Township (9) Trabzon (46) rma (26) Baki (195) Kampung Kasing (2) Couman (4) Desa Lenangguar (2) Parc National de Waza (2) Longstaff Peaks (3) City of Little Sioux (20) East Chattanooga (62) Punta Avalo (3) Ban Tham (24) Bacu (17) South Atlantic Ocean (13) Little Axe Independent School District (5) Precious Peaks (4) Casual Branch (19) Kyriad Limoges Sud
Continued on next page

(2) Pirsaat (23) Goderich Airport (11) Krasnyye Baki (23) Twin Peaks (42) Town of Lake Hamilton (15) Little York Township (19) Goli Rid (21) Lenkiyio (2) Parc Forestier de Hann (2) Town of Greece (31) DallasFort Worth International Airport (2) South Glamorgan (63) Jabal Tali (7) South Sikkim (21) Chowasokwe (4) Ban (238) Mufazat al Mawt (5) Golden Valley County (19) Qala (76) Corners (190) Kampung Ulu Tiram (5) Zonhoven (17) Comuna Tetoiu (5) Paret River (2) Chak Four (17) Tienen (41) Medina (91) Halethorpe (23) Shenley Church End (20) Chovakaranga (2) Friendly Corners (17) Rs Iouk (19) Fatick Department (20) Punta (237) Kalbuh Park (11) City of Crowley (20) Willowdale (87) Pulau Babi Besar (3) Jawf al Abd (6) Giurgiu (49) Grand Prairie Municipal Airport (14) Kf (74) Jebel Oued en Nemeur (16) Webers Peaks (5) School Lake (22) Cap Gros (46) Besar (171) Chk (11) Ban Ti Te (3) Town of Gorham (23) Gole (199) Calvary (57) Laguna Gaiba (8) Tr (236) Church Pond (34) Life Ambulance (20) Samsun (51) Hamilton (222) Makassar (28) BeuzecCapSizun (10) Crescent Reserve (69) Lac Gris (7) Erzegovina (11) Tom Bayou (27) Park historical Township (19) Tom Price (22) Cypress Ridge Township (3) Amphitheatre Peaks (4) Osanippa (7) Flannigan Corners (4) Davey Point (12) Little Blue Township (15)

Laguna Suches (6) Black RiverMatheson (17) Sahqaya (3) Shaumyanovskiy Rayon (22) Borups Corners (3) Court (220) Punta Caleta Larga (2) Bailly (62) Kester Peaks (2) Rock Township (45) Grants Camps (16) Ban Ta Luang (11) Oval (154) Carmel

Hamlet (22) Sagami Bank (22) Mersin (47) North Fork State Wildlife Refuge (14) Q Chk (30) Pulau Merak Besar (2) Town of Cumberland (18) Summit School Airport (13) Bailey Corners (27) Town of Gray (21) Golden (222) Crater Lake (47) Sandy Point District (16) Church Fenton (32) Soholt Peaks (6) Uchastok Sok (18) Crutch Peaks (6) Buon Ma Thuot (16) Church Lakes (28) Chak Nine (19) Dorylaeum (6) Urochishche (118) Rock Creek Township (45) La Grecia (10) City of Zenda (6) Tiko Airport (6) Maple Ridge Township (29) Yarra Reserve (24) Mouscron (28) Kolda Department (20) Eagle Township (50) Puerto Rico Ridge (41) Tehuantepec Ridge (6) Baa Orion (21) Ras Chani (12) Yozgat (48) Tlyadal (8) Torch River No 488 (13) Bodega Head (13) Mikhaylovskiy Uchastok (26) Tom Bean (24) Province of Laguna (17) Ro (237) Farrell Corners (11) Parc National des Virunga (4) Lake Killarney (49) Casual (29) Hamlet Park (19) Le Cirand (5) Gole Strane (4) historical (139) Village of Hales

Corners (2) Borough of Audubon Park (11) East Rock Bluff Township (2) Lac NoirGris (23) Kyriad Avignon (8) Ambulance (121) Oil Trough Township (2) Little Salt Township (12) Borough of Park Ridge (14) Rock (222) Desa Tapaan (2) City of Little River (12) Springfield Ambulance (33) Kampung Kuala Wau (8) Little Valley Township (9) Church Shocklach

(14) Palawan Trough (3) Kesteloots Trailer Court (2) Sorapa (7) Sakk (7) Kampung Pasir Gudang Baru (2) Smith Lake (40) Lamia (78) Square Deal Hill (4) Bassi Department (39) Current River Township (20) Coulee State Wildlife Refuge (15) Verkhnyaya Pkhiya (21) Little San Salvador (4) Manipur South (33) City of Sansom Park (8) City of Hurst (17) Wairarapa South County (3) Tokat (35) Bures Hamlet (28) Old River Township (20) Campbell Corners (22) Zapadnoye Lake (8) Crique La Villette (4) Huntington Park (37) City of Blue Mound (27) Arrondissement Turnhout (23) Kampung Mangsuk (5) Pkhovo (18) Lake Tom (54) Town of Little Wolf (15) Royals Court (17) Rundkino (2) Pirsaat Burnu (2) Kecamatan Sawah Besar (2) Salt Rock Township (19) Chaplin Oval (13) City of Demopolis (10) Sango Point (5) South Shields (32) City of North Little Rock (3) Judeul Timi (7) Ban To Mo (30) School (191) Hamlet (194) East Hamilton Township (18) Chak Seventeen (10) Leiman (11) Vasilyevka (23) Jawf (128) HauteBanio Department (4) Sulu Trough (7) Rago (50) Desa Bengkak (2) Park Township (43) Blue Ridge Township (22) Gol (73) Greenup Township (22) Kampung Sedenak (8) Bonney Lake (30) Gor (137) Hales

Continued on next page

Corners (21) Comuna Valea Viilor (3) Northwest Ambulance District (8) Porongas (4) Kampung Panchor (21) Ouindigui Department (4) La Villette (11) Golden Lake (42) Rock Falls Township (19) Cap Foreland (5) Woodhaven Court (20) Ban Phueng (24) Bodega Canyon (5) Hanover Park (42) Turnhout (25) Parc Ahuntsic (19) Little School Lot Lake (9) Punta Arena (46) Lake Hamilton (52) Hobe Sound National Wildlife Refuge (8) Arrondissement de Lascahobas (4) Atlantique Department (24) Refuge Point (20) Kyriad (101) Kampung Gantuk (3) Lope Department (12) Brunswick Golden Isles Airport (8) Camberwell South (23) Bodega Azul (42) City of Hamlet (19) Ogoulou Department (3) Bodega Pampa (13) Sechenovo (10) Lake Bemba (6) Marmara Trough (9) Rio Saliceto (22) Umm Ghuayyah

(22) Menemen (24) Cerro Bodega (4) Ban Palian (11) Lolo Bouenguidi Department (5) Parc Infantil (4) Kovalam Point (4) Corral Peaks (9) Round Rock (46) SintTruiden (5) Parc Anderson (8) Park (237) Ban Tamot (4) Bennett Township (41) Golden City Township (11) Parc (222) Mossey River (20) Ridge (208) Magnier Peaks (4) Pkhiya (15) Ro Tercero (7) Webster Peaks (11) Forest Hill (69) River (238) Desa Karangrejo (29) Vilvoorde (40) Tri Ro (5) Croteau (39) Golden Lake Township (19) Urochishche Kharkovka (6) Kampung Peradong (3) Kampung Pondoi (5) Eldoinyo (17) Punta Cana (46) Zelzate (22) Airport (184) City of New

Cumberland (18) Knysna (46) Punta Tejupan (3) Laguna Mapache (2) School Pond (33) Turtle River No 469 (7) Kampung Seelong (2) Marengo (198) Jawf al Athlah (3) Al azm (14) City of Grosse Pointe Park (12) la Ciutadella (16) Lake Paoay (3) Punta Morillo (6) Yli (174) Ban Tom Klang (34) Kyriad Grenoble Seyssins (3) Hakkari (25) Dolgi Rid (6) City of Watauga (18) Burks Corners (3) Maltosrova (2) Kyriad Gap (21) South Dublin (40) Tal Chl (21) Reserve (184) Culpeo (17) Godfreys Reserve (4) Desa Mojotengah (4) Oula Department (24) City of Westworth Village (6) Ban Talat Bueng (19) Kampung Sungai Miang (2) Tinley Park (26) Crvenorovinski (2) Porphyry Peaks (5) Town of Orange Park (17) Ban Phai (47) Hulme (66) West Deal (12) Rock Pile Peaks (4) Makwana School (2)

Town of Providence (27) Greenleys Corners (3) Desa Babadan (14) Lac Tom (32) Chojlloni (2) Deal Lake (17) Cumberland (234) Coyote Peaks (15) Atakora Department (22) Lake Bangweulu (20) Polevskoy Uchastok (2) Stan Sar (4) Bottou Department (2) Wd Sidr (15) Pulau Redang Airport (5) Lake Ruko (3) Isle of Wight Department of Natural Resources Management Area (4) Seraing (34) Bullocks Corners (11) Rosso Department (21) Robert Cape (41) Baldwin Park (51) Slatina (105) Uchastok (116) Baudin Peaks (3) Sagami Canyon (21) Pizozerka (2) Martin Rid (6) Borough of Shiremanstown (2) Torhout (23) City of Hamilton (9) Wuro Gole (2) Papago Hamlet (5) Rock Island Township (18) Samsun Ridge (2) Chak (237) Kampung Rial (14) Plateaux Department (18) Villette (187) Jabal Umm Raqabah (2) Desa Sokawera (3) Punta Caleta (4) Desa Wunung (2) Deal (192) South Big Rock Township (11) Haughton Court (20) Desa Manggis (21)

Sunda Trough (13) Taleh (11) AugustaMargaret River Shire (22) Aydin (36) Rivera Peaks (11) Mys Sangachal (2) Lake Charles (50) Talle Shaghl (11) Church Mesa (27) Three Corners Lake (6) City of Saginaw (18) Bakonya (13) Punta Bayas (10) Murdock (59) Soraoco (2) Mougoutsi Department (3) Town of Hamilton (14) Bodega Marine Reserve (4) Ban Mai (35)

Tom Peter Lake (16) Kampung Gadek (6) Kampung Baharu (43) Bakony (58) Pettifor (2) Hypaepa (3) Erzincan (40) K Orra (11) Town of Casco (24) Cale Oval (15) Lone Tree Township (43) City of Little Rock (19) Col NotreDame (22) Ro Coco (5) Oued Lill (5) Stormy Peaks (2) Ban Nong Muang (70) Bukit Besar Jelai (4) Wallula (38) Qarada (37) Merklings Trailer Court (2) Sang Kan (3) Lystad Bay (48) Libya (56) Al Bb (23) Formosa do Rio Preto (12) Ban Phalai (9) Bishop Corners (23) Wintercone (3) Laguna del Barn (3) Marengo Township (54) Cleveland (97) Soudougui Department (5) Court Park (38) Sudr (4) City of Golden Valley (8) Golden Pond (19) Cap (222) Lake County (36) Ibadan Airport (9) West Norriton Ambulance (3) Diyak (6) Bakonyalja (2) Little Deal Island (9) Dr Dirang (3) Department of Transportation (21) Tournai (46) Grant Hill (35) Town of Little Mountain (20) Zemli (9) Morocco (51) Hatien (3) Evangelistical (2) Violeta (93) Mayo River District (19) Providence (98) Aitken Cove (20) Camps (189) McKinleyville (22) Nilombot Golden (4) Creswick Peaks (5) Pilot Rock Township (21) City of Forest Hill (12) Kingman (194) Rio nellElba (21) Oued Laou (5) Sredni Rid (4) Comuna Zeme (8) Le Donjon (24) Piatra Neam (20) Shindi School (9) Soranpampa (2) Rho (97) Kampung Petuh (5) Al Huff (18) Town of Pound Ridge (16) Little River Township (6) Markov Rid (6) Ban Phan Don (22) Farleys Corners (6) White Township (48) Yellowstone National Park (18) Borough of Little

Continued on next page

Meadows (11) Tarrant Monkton (13) Masai Mara Game Reserve (3) Rio (221) Troms Airport (21) Yellowstone National Park County historical (14) Borough of Prospect Park (20) City of Linden (27) Claremont Oval (38) Burch Peaks (8) Belmont Center Gas Field (3) Commissioner District 6 (5) Rock County (16) Commissioner District 7 (7) Commissioner District 8 (34) Commissioner District 9 (8) Church Basin (31) Commissioner District 5 (29) Rid (218) Al Bay (30) Durian Besar (41) Ban Phaeo (23) Ridge Peak (52) Laguna Tequesquitengo (3) Greece (80) Bone Bay (36) Shiloh (77) Soloviu (2) Blue Mound (63) Ras (235) Oued Chelif (3) Court Square (24) Caleta Olivia (15) Mpassa Department (6) South River District (12) Town of Little Suamico (8) Ras Buur Gaabo (2) Pizya (4) Parsley Swamp (18) Hamilton historical Township (18) Four Corners (55) Chilchi (17) Paili Plantation (19) Porter Corners (21) Refuge Key (21) Tongeren (46)

Ras Tenewi (2) Mons (99) Millennium Park (26) Bodega Dunes (13) Golden Township (21) Ro Cuarto (9) Mys Alyat (20) Neptunes Window (2) Town of Freeport (16) Park District (41) Advance Ambulance (27) Oued (234) Tribhuwan Airport (3) Tli (23) Henry Township (57) Belek (28) Solberg Inlet (23) Department (130) Durham Park Township (19) Basse-Banio Department (49) Nagh Bak (20) Oval Lake (24) South Kivu (25) Cottonwood Township (35) Kin Uly (31) Les Petits Camps (16) City of Keller (19) Ylivieska Airport (4) Smith Ambulance Service (19) Iris Refuge Island (14) Greens Corners (21) Ban Phon Ngam (52) Unorganized

Territory of Kingman (18) Umm Qulayah (3) Ro Gallegos (4) Cap Aubert (18) North River District (29) Ban Chun Luang (3) Cerro Sorapa (20) kyriad sud (22) Sweet Water (44) Little Deep Township (5) Tom Thomson Lake (13) Erzurum (45) Ramana (52) Caleta (176) Bodega Bay (8) Ban Khrai (23) Geneva (97) Tom (220) Mount Tom Lake (13) Pleasant Ridge historical Township (4) Bonin Trough (8) Baltics Corners (2) Refuge (162) Senglea Point (5) Hamlet Farms (10) Church Pine Lake (15) Kyriad Deauville St Arnoult (3) Rysa (14) Dsa (15) Desa (198) Donjon (75) Kampung Ayer Keroh (17) Hotamville (5) Doonside Oval (7) Aristovo (29) Tali (220) Mibzal (17) Oval Peak (5) Rio Papuri (6) Kyriad Rimini (5) Manggar Besar (4) Virgin Islands Trough (6) South 24 Parganas (19) Zooks (21) Kastamonu (45) Puta (41) Batikent (5) Kampung Taburan Besar (3) Tiderishi (2) Gamla stan (31) Kampung Cenering (2) New Church (44) Magherabuoy (2) City of Dalworthington Gardens (2) Arrondissement (189) Rava Point (6) Oued Ifrane (11) Cerro Largo (58) Parsley (60) Desa Krandon (6) Petit lac Tom (4) Kny (102) Dixons Mills (17) Borough of Deal (13) Lomonosov Ridge (8) Cream Ridge Township (5) Caleta Chica (23) School Section Lake (31) City of Forest Park (13) Cp Tin (27) Hampton Court (35) Parc Aldred (3) Kortrijk (46) Tipperary South Riding (8) Refuge Pond (61) City of Church Hill (5) Lake (235) Tal Bolgh (2) Jubail (25) Community Ambulance (53) Urochishche Bannikovo (6) Everglades National Park (23) Holth Peaks (2) Caleta Mangle (6) Little Tom Lake (10) Raas Kaambooni (4) Cap Coster (4) Ban Bueng Bon (6) Emma Cove (7) Arrondissement Tielt (18) Portsmouth Ambulance (38) Le Cap (29) South (237) Borough of Shippensburg (9) Zamboanga (53) Hasvik Airport (6) Jadwal (112) Bodega Island (7) Gain Stan (8) Slieveanorra (4) Crestone Peaks (13) Siillaviit (4) Marengo Lake (15) Gole Khel (39) Kingman Gulch (11) Puu Greci (15) Dialgaye Department (4) Church Hill (49) Cheadle Hulme Railway Station (17) Little Black Township (17) Nashville (79) Cerralvo Trough (2) Sagami Trough (4) Campbells Corners (25) Hamlet Lake (18)

Ellenbogen (43), Torre (151), Dagny (25), Dolgoff (9), Fred (220), Crawley (68), Richard (238), Mustafa (145), Rongomai (11), Harry (194), Rogers (178), Julius (185), Rocque (20), Blaeser (13), Korkmaz (32), Robyn (119), Phife (22), Trish (103), James (240), Aspasius (7), Jean (192), Reddy (140), Isaac (187), Karl (206), Wickware (9), Woodforde (46), Tomter (10), Dicki (32), Hamilton (199), Keller (161), Masango (6), Parizeau (24), Kessler (120), trento (1), Eugene (199), Di (174), Madagascar (37), Acheamphong (4), Barigozzi (5), Freddy (147), Gortz (25), Menon (100), Seiichi (46), Victor (186), Gayifi (4), Ferrero (68), Deo (93), Andrew (214), Jenkins (165), Powell (187), Barteczko (5), Samuel (48), Fitzsimons (61), null (23), Konee (20), Trubetsky (23), Chtiba (1), Goulet (60), Barber (165), Rene (173), Boyer (121), Mackintosh (84), Francisco (193), Petre (89), Rupert (145), Aubelin (2), Spinola (30), Golden (161), Arthur (216), John (119), Cory (163), Percival (118), Campbell (217), Heard (115), Jim (196), Bristow (79), Tony (178), Oswaldo (83), Robbemond (2), Forster (122), George (114), Wilber (94), Fingar (11), Frank (221), Adam (193), Stewart (211), Agger (49), Gerhard (151), Hunt (175), Ratner (29), Bonaparte (71), Green (219), Oeyvind (1), Pons (75), William (295), Peter (309), Israel (168), Argiris (15), Gourault (2), Dad (110), Basile (91), Elrod (53), Cvijanovic (6), Valle (157), Scott (188), Leman (83), Anders (172), Rebecca (155), Morgan (213), Dauncey (25), Takeda (69), Vanvelthoven (6), Marianna (77), Raymond (192), Ezekias (7), Mincu (21), Shandruk (8), Benjamin (215), Wang (151), Brouard (16), Nathaniel (217), Safdar (50), Elliott (180), Carl (203), Zhuo (29), Ian (176), Don (135)

Table B.1: Queries used to collect samples of Person descriptions: query (nr of samples retrieved)

Acadians (91), Press (235), Funding (222), stations (243), French (275), Jew (117), Schell (136), University (309), Centre (341), India (237), Holden (235), Niels (184), Records (242), Champaign (261), Vanity Project The (4), Urbana (244), Orchestral Manoeuvres in the Dark (24), Coral (309), Heavy Into Jeff (29), School (306), Shimbun (49), Current (258), Atomic Kittens (20), Transylvanian (45), Francis (341), Saint (297), Party (278), Parsons Jerry Blue Jeans The (7), Socialist (166), Left (177), Latin (268), The Unknown Project (8), Brazilian (250), Switchblade Kittens (7), Muckleshoot (12), Environment (240), Jerry Parsons The Blue Jeans (21), Basics (234), Im (226), Ltd (243), 2 in da Bush (22), Tunnel Allstars (27), Tilburg (112), Chicago (350), Newcastle (283), Unknown Project The (16), Dharma (218), Sense (214), Entertainment (333), Endorsement (55), Vanderbilt (230), Mirrors (165), Anathema (34), Devi (145), The Alan Parsons Project (24), Earth (336), British (341), Paulista (209), The Octopus Project (26), Hello (149), Darth (61), Nets (244), Unknown The (67), The Unknown (40), Production (284), Basilicata (165), Radiohead (25), German (237), Americans (311)

Table B.3: Queries used to collect samples of Organization descriptions: query (nr of samples retrieved)

Bibliography

[1] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *J. Mach. Learn. Res.*, 1:113–141, September 2001.

[2] G Antoniou and F. van Harmelen. *A Semantic Web Primer.* MIT Press, 2004.

[3] Rohan Baxter, Peter Christen, and Tim Churches. A comparison of fast blocking methods for record linkage. In *KDD 2003 WORKSHOPS*, pages 25–27, 2003.

[4] B. Bazzanella, P. Bouquet, and H. Stoermer. A cognitive contribution to entity representation and matching. Technical report, (DISI-09-004) University of Trento, 2009.

[5] B. Bazzanella, H. Stoermer, and P. Bouquet. A bayesian model for entity type disambiguation. In *Artificial Intelligence: Methodology, Systems, and Applications*, pages 121–130. Springer Berlin / Heidelberg, 2010.

[6] Barbara Bazzanella. *Uniqueness in Cognition.* PhD thesis, Doctoral School in Physicological and Education, 2010.

[7] Barbara Bazzanella, Paolo Bouquet, and Heiko Stoermer. Top level categories and attributes for entity representation. Technical report, University of Trento, 2008.

[8] Barbara Bazzanella, Junaid Ahsenali Chaudhry, Themis Palpanas, and Heiko Stoermer. Towards a general entity representation model. In *SWAP*, 2008.

[9] Ron Bekkerman and Andrew McCallum. Disambiguating web appearances of people in a social network. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 463–470, New York, NY, USA, 2005. ACM.

[10] O. Benjelloun, H. Garcia-Molina, D. Menestrina, Qi Su, S. Whang, and J. Widom. Swoosh: a generic approach to entity resolution. *The VLDB Journal*, 18:255276, 2009.

[11] T. Berners-Lee. What do http uris identify? http://www.w3.org/DesignIssues/HTTP-URI, January 2007. Discussion started in the July 2002.

[12] T. Berners-Lee, J. A. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May, 2001. http://www.sciam.com/2001/0501issue/0501berners-lee.html.

[13] Tim Berners-Lee. Design Issues – Linked Data. Published online, May 2007. `http://www.w3.org/DesignIssues/LinkedData.html`.

[14] Tim Berners-Lee. Giant global graph. *Decentralized Information Group - http://dig.csail.mit.edu/breadcrumbs/node/215*, 2007.

[15] I. Bhattacharya and L. Getoor. Iterative record linkage for cleaning and integration. In *DMKD '04: Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 11–18, New York, NY, USA, 2004. ACM.

[16] I. Bhattacharya and L. Getoor. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1, March 2007.

[17] I. Bhattacharya and L. Getoor. Query-time entity resolution. *Journal of Artificial Intelligence Research*, 30:621–657, December 2007.

[18] Mikhail Bilenko, Raymond Mooney, William Cohen, Pradeep Ravikumar, and Stephen Fienberg. Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23, September 2003.

[19] Mikhail Bilenko and Raymond J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 39–48, New York, NY, USA, 2003. ACM.

[20] C. Bizer, R. Cyganiak, and T. Heath. How to publish linked data on the web. online tutorial, July 2007.

[21] Jr. Bois, N. S. D'Andrea Du. A solution to the problem of linking multivariate documents. *Journal of the American Statistical Association*, 64(325):pp. 163–174, 1969.

[22] David Booth. Denotation as a two-step mapping in semantic web architecture. In *Identity and Reference in web-based Knowledge Representation (IR-KR2009)*, 2009.

[23] S. Bortoli, P. Bouquet, H. Stoermer, and H. Wache. Foaf-o-matic - solving the identity problem in the foaf network. In *Proceeding of SWAP 2007 - Fourth Italian Semantic Web Workshop (To Appear)*, 2007.

[24] P. Bouquet, C. Ghidini, and L. Serafini. Identity and reference on the global giant graph. In *"Identity and Reference in web-based Knowledge Representation" (IR-KR2009 at IJCAI-09)*, 2009.

[25] P. Bouquet and H. Stoermer. OKKAM: Enabling an Entity Name System for the Semantic Web. In *Proceedings of the I'ESA2008 Workshop on Semantic Interoperability*, 2008. to appear.

[26] P. Bouquet, H. Stoermer, and B. Bazzanella. An Entity Naming System for the Semantic Web. In *Proceedings of the 5th European Semantic Web Conference (ESWC2008)*, LNCS, 2008. to appear.

[27] P. Bouquet, H. Stoermer, D Cordioli, and G. Tummarello. An Entity Name System for Linking Semantic Web Data. In *Proceedings of LDOW2008*, April 2008. `http://events.linkeddata.org/ldow2008/papers/23-bouquet-stoermer-entity-name-system.pdf`.

[28] P. Bouquet, H. Stoermer, C. Niederee, and A. Mana. Entity Name System: The Backbone of an Open and Scalable Web of Data. In *Proceedings of the IEEE International Conference on Semantic Computing, ICSC 2008*, number CSS-ICSC 2008-4-28-25, pages 554–561. IEEE Computer Society, August 2008.

[29] Paolo Bouquet, Fausto Giunchiglia, Frank Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. C-owl: Contextualizing ontologies. In Dieter Fensel, Katia Sycara, and John Mylopoulos, editors, *The Semantic Web - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 164–179. Springer Berlin Heidelberg, 2003.

[30] Paolo Bouquet, Themis Palpanas, Heiko Stoermer, and Massimiliano Vignolo. A conceptual model for a web-scale entity name system. In *Asian Semantic Web Conference (ASWC)*, Shanghai, China, 2009.

[31] Daren C. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence: The International Journal of Research into New Media Technologies*, 14(1):75–90, 2008.

[32] D. G. Brizan and A. Tansel. A survey of entity resolution and record linkage methodologies. *Communication of IIMA*, 6(3):41–50, 2006.

[33] Horacio Camacho and Abdellah Salhi. A redundancy detection approach to mining bioinformatics data. *Computer Aided Methods in Optimal Design and Operations*, 7:89–98, 2006.

[34] Greg Carlson. Thematic roles and the individuation of events. In Susan Rothstein, editor, *Events and Grammar*, pages 35–51. Kluwer Academic Publishers, 1998.

[35] M. Carrara, P. Giaretta, V. Morato, M. Soavi, and G. Spolaore. Identity and modality in ontoclean. In A Varzi and L. Vieu, editors, *Formal Ontology in Information Systems*. IOS Press, 2004.

[36] Massimiliano Carrara and Pieter E. Vermaas. The fine-grained metaphysics of artifactual and biological functional kinds. *Synthese*, 169(1):125–143, July 2009.

[37] S. Castano, A. Ferrara, S. Montanelli, and D. Lorusso. Instance matching for ontology population. In *Proc. of the 16th Italian Symposium on Advanced Database Systems*, pages 22–25, 2008.

[38] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.*, 26(2):4:1–4:26, June 2008.

[39] S. Chaudhuri, A. Das Sarma, V. Ganti, and R. Kaushik. Leveraging aggregate constraints for deduplication. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 437–448, New York, NY, USA, 2007. ACM.

[40] S. Chaudhuri, V. Ganti, and R. Motwani. Robust identification of fuzzy duplicates. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 865 – 876, 2005.

[41] Jie Chen, Cheqing Jin, Rong Zhang, and Aoying Zhou. A learning method for entity matching. In *Proceedings of 10th International Workshop on Quality in Databases QLDB 2012*, 2012.

[42] Z. Chen, D. Kalashnikov, and S. Mehrotra. Exploiting relationships for object consolidation. In *Proceedings of the 2nd international workshop on Information quality in information systems*, IQIS '05, pages 47–58, New York, NY, USA, 2005. ACM.

[43] Tim Churches and Peter Christen. Some methods for blindfolded record linkage. *BMC Medical Informatics and Decision Making*, 4, 2004.

[44] K. G. Clark. Identity crisis, September 11 2002.

[45] Carol Cleland. On the individuation of events. In *Synthese*, volume 86, pages 229–254. Springer, 1991.

[46] William W. Cohen, Henry Kautz, and David McAllester. Hardening soft information sources. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 255–259, New York, NY, USA, 2000. ACM.

[47] William W. Cohen, Pradeep D. Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, pages 73–78, 2003.

[48] David Cohn, Richard Ladner, and Alex Waibel. Improving generalization with active learning. In *Machine Learning*, pages 201–221, 1994.

[49] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[50] P. Cudré-Mauroux, P. Haghani, M. Jost, K. Aberer, and H. De Meer. idmesh: graph-based disambiguation of linked data. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 591–600, New York, NY, USA, 2009. ACM.

[51] Donald Davidson. The individuation of events. In *Essays on Actions and events.* Claredon Press, 1980.

[52] M.G. de Carvalho, A.H.F. Laender, M.A. Goncalves, and A.S. da Silva. A genetic programming approach to record deduplication. *Knowledge and Data Engineering, IEEE Transactions on*, 24(3):399 –412, march 2012.

[53] Timothy de Vries, Hui Ke, Sanjay Chawla, and Peter Christen. Robust record linkage blocking using suffix arrays. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 305–314, New York, NY, USA, 2009. ACM.

[54] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[55] D. Dey, S. Sarkar, and P. De. Entity matching in heterogeneous databases: a distance-based decision model. In *System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on*, volume 7, pages 305 –313 vol.7, jan 1998.

[56] Elena Deza and Michel Marie Deza. *Encyclopedia of Distances.* Springer, 2013.

[57] A. Doan, Y. Lu, Y. Lee, and J. Han. Profile-based object matching for information integration. *Intelligent Systems, IEEE*, 18(5):54 – 59, 2003.

[58] A. Elmagarmid, P. Ipeirotis, and V. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19:1–16, 2007.

[59] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching.* Springer-Verlag, Heidelberg (DE), 2007.

[60] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64:pp. 1183–1210, 1969.

[61] J.; Richardson T. Ganesh, M.; Srivastava. Mining entity-identification rules for database integration. In *Proceedings of the Second International Conference on Data Mining and Knowledge Discovery*, 1996.

[62] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with dolce. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, EKAW '02, pages 166–181, London, UK, UK, 2002. Springer-Verlag.

[63] Aldo Gangemi and Valentina Presutti. Towards an OWL Ontology for Identity on the Web. In *Semantic Web Applications and Perspectives (SWAP2006)*, 2006.

[64] Aldo Gangemi and Valentina Presutti. A grounded ontology for identity and reference of web resources. In *i3: Identity, Identifiers, Identification. Proceedings of the WWW2007 Workshop on Entity-Centric Approaches to Information and Knowledge Management on the Web, Banff, Canada, May 8, 2007.*, 2007.

[65] L.E. Gill. Ox-link: The oxford medical record linkage system. In *Proceedigs of Record Linkage Workshop and Exposition*, 1987.

[66] H. Glaser, A. Jaffri, and I. Millard. Managing co-reference on the semantic web. In *WWW2009 Workshop: Linked Data on the Web (LDOW2009)*, April 2009.

[67] K. Goiser and P. Christen. Towards automated record linkage. In *Proceedings of the fifth Australasian conference on Data mining and analystics - Volume 61*, AusDM '06, pages 23–31, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[68] N. Guarino and C. Welty. Identity, unity and individuality: Towards a formal toolkit for ontological analysis. In *ECAI-2000: The European Conference on Artificial Intelligence*, 2000.

[69] Nicola Guarino. The role of identity conditions in ontology design. In *Proceedings of the International Conference on Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science*, COSIT '99, pages 221–234, London, UK, UK, 1999. Springer-Verlag.

[70] Nicola Guarino and Chris Welty. An overview of ontoclean. In Steffen Staab and Rudi Studer, editors, *The Handbook on Ontologies*, pages 151–172. Springer-Verlag, 2004.

[71] Sudipto Guha, Nick Koudas, Amit Marathe, and Divesh Srivastava. Merging the results of approximate match operations. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 636–647. VLDB Endowment, 2004.

[72] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46(1-3):389–422, March 2002.

[73] Harry Halpin, Patrick Hayes, James McCusker, Deborah McGuinness, and Henry Thompson. When owl:sameas isnt the same: An analysis of identity in linked data. In Peter Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Pan, Ian Horrocks, and Birte Glimm, editors, *The Semantic Web ISWC 2010*, volume 6496 of *Lecture Notes in Computer Science*, pages 305–320. Springer Berlin / Heidelberg, 2010.

[74] Harry Halpin and Patrick J. Hayes. When owl:sameAs isn't the Same: An Analysis of Identity Links on the Semantic Web. In *Workshop on Linked Data on the Web*, April 2010.

[75] P J Hayes and H Halpin. In defense of ambiguity. *International Journal on Semantic Web and Information Systems.*, Vol. 4(2):1–18, Apr.-June 2008.

[76] S. Axford H.B. Newcombe, J.M. Kennedy and A. James. Automatic linkage of vital records. *Science*, 130:954–959, 1959.

[77] M.A. Hearst, S.T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18–28, 1998.

[78] R. Hecht and S. Jablonski. Nosql evaluation: A use case oriented survey. In *Cloud and Service Computing (CSC), 2011 International Conference on*, pages 336 –341, dec. 2011.

[79] M. A. Hernández and S. Stolfo. The merge/purge problem for large databases. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, SIGMOD '95, pages 127–138, New York, NY, USA, 1995. ACM.

[80] Mauricio A. Hernández and Salvatore J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Min. Knowl. Discov.*, 2(1):9–37, January 1998.

[81] A. Hogan, A. Polleres, J. Umbrich, and A. Zimmermann. Some entities are more equal than others: statistical methods to consolidate linked data. In *Proceedings of the Workshop on New Forms of Reasoning for the Semantic Web: Scalable & Dynamic (NeFoRS2010)*, 2010.

[82] Aidan Hogan, Andreas Harth, and Stefan Decker. Performing object consolidation on the semantic web data graph. In *i3: Identity, Identifiers, Identification. Proceedings of the WWW2007 Workshop on Entity-Centric Approaches to Information and Knowledge Management on the Web, Banff, Canada, May 8, 2007.*, 2007.

[83] Chih-Chung; Hsu, Chih-Wei; Chang and Chih-Jen Lin. A practical guide to support vector classification. Available online., 2003.

[84] Wei Hu, Jianfeng Chen, and Yuzhong Qu. A self-training approach for resolving object coreference on the semantic web. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 87–96, New York, NY, USA, 2011. ACM.

[85] Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is np-complete. *Inf. Process. Lett.*, 5(1):15–17, 1976.

[86] Ekaterini Ioannou, Claudia Niederée, and Wolfgang Nejdl. Probabilistic entity linkage for heterogeneous information spaces. In *Proceedings of the 20th international conference on Advanced Information Systems Engineering*, CAiSE '08, pages 556–570, Berlin, Heidelberg, 2008. Springer-Verlag.

[87] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 64–67, New York, NY, USA, 2010. ACM.

[88] Robert Isele and Christian Bizer. Learning expressive linkage rules using genetic programming. *Proc. VLDB Endow.*, 5(11):1638–1649, July 2012.

[89] M.A. Jaro. Unimatch: A record linkage system: Users manual. Technical report, US Bureau of the Census, 1976.

[90] Trond Grenager Jenny Rose Finkel and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, 2005.

[91] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural Comput.*, 13(3):637–649, March 2001.

[92] Aleksandar Kellenberg. Identifying criteria of identity. *Ontology Metaphysics*, 10:109–122, 2009.

[93] W. Kent. The breakdown of the information model in multi-database systems. *ACM SIGMOD Record*, 20(4):10–15, 1991.

[94] W. Kent. The Unsolvable Identity Problem. In *Extreme Markup Languages*, 2003.

[95] W. Kent, R. Ahmed, J. Albert, M. Ketabchi, and M. Shan. Object identification in multidatabase systems. In *Proceedings of the IFIP WG 2.6 Database Semantics Conference on Interoperable Database Systems (DS-5), Lorne, Victoria, Australia, 16-20 November 1992*, pages 313–330, 1992.

[96] R. Kimball and J. Caserta. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data.* John Wiley & Sons, 2004.

[97] S. C. Kleene. *Introduction to Metamathematics.* Princeton, 1950.

[98] Grzegorz Kondrak. N-gram similarity and distance. In *Proceedings of Twelfth Intl Conf. on String Processing and Information Retrieval*, 2005.

[99] Hanna Köpcke and Erhard Rahm. Training selection for tuning entity matching. In *QDB/MUD*, pages 3–12, 2008.

[100] Hanna Köpcke, Andreas Thor, and Erhard Rahm. Evaluation of entity resolution approaches on real-world match problems. *Proc. VLDB Endow.*, 3(1-2):484–493, September 2010.

[101] Saul Kripke. *Naming and Necessity.* Basil Blackwell, Boston, 1980.

[102] Vladimir Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–10, 1966.

[103] E.-P. Lim, J. Srivastava, S. Prabhakar, and J. Richardson. Entity identification in database integration. In *Proceedings. Ninth International Conference on Data Engineering, 1993.*, pages 294 –301, apr. 1993.

[104] E. J. Lowe. *Kinds of Being. A Study of Individuation, Identity, and the Logic of Sortal Terms.* Blackwell, 1989.

[105] E. J. Lowe. What is a criterion of identity? *The Philosphical Quarterly*, 39:1–21, 1989.

[106] Geoffrey Holmes Bernhard Pfahringer Peter Reutemann Ian H. Witten Mark Hall, Eibe Frank. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[107] Cynthia Matuszek, John Cabral, Michael Witbrock, and John Deoliveira. An introduction to the syntax and content of cyc. In *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, pages 44–49, 2006.

[108] M. Michalowski, S. Thakkar, and C. Knoblock. Exploiting secondary sources for automatic object consolidation. In *In Proceedings of the ACM SIGKDD-03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, 2003.

[109] M. Michalowski, S. Thakkar, and C. Knoblock. Exploiting secondary sources for unsupervised record linkage. In *In IIWeb*, 2004.

[110] Matthew Michelson and Craig A. Knoblock. Learning blocking schemes for record linkage. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, 2006.

[111] Zoltán Miklós, Nicolas Bonvin, Paolo Bouquet, Michele Catasta, Daniele Cordioli, Peter Fankhauser, Julien Gaugaz, Ekaterini Ioannou, Hristo Koshutanski, Antonio Maña, Claudia Niederée, Themis Palpanas, and Heiko Stoermer. From web data to entities and back. In *Proceedings of the 22nd international conference on Advanced information systems engineering*, CAiSE'10, pages 302–316, Berlin, Heidelberg, 2010. Springer-Verlag.

[112] S. Minton, C. Nanjo, C. Knoblock, M. Michalowski, and M. Michelson. A heterogeneous field matching method for record linkage. *Data Mining, IEEE International Conference on*, 0:314–321, 2005.

[113] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[114] Alvaro Monge and Charles Elkan. The field matching problem: Algorithms and applications. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[115] Sreerama K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Min. Knowl. Discov.*, 2(4):345–389, December 1998.

[116] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:44353, 1970.

[117] Axel-Cyrille Ngonga Ngomo. On link discovery using a hybrid approach. *J. Data Semantics*, 1(4):203–217, 2012.

[118] Xing Niu, Shu Rong, Haofen Wang, and Yong Yu. An effective rule miner for instance matching in a web of data. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 1085–1094, New York, NY, USA, 2012. ACM.

[119] Lawrence Philips. Hanging on the metaphone. *Computer Language*, 7, 1990.

[120] John C. Platt. Advances in kernel methods. chapter Fast training of support vector machines using sequential minimal optimization, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[121] W. V. Quine. Events and reification. In E. Lepore & B. McLaughlin (eds.), editor, *Actions and Events: Perspectives on the Philosophy of Davidson.* Blackwell, 1985.

[122] Willard Quine. *Ontological Relativity and Other Essays.* Columbia University Press, 1969.

[123] J. R. Quinlan. Generating production rules from decision trees. In *Proceedings of the 10th international joint conference on Artificial intelligence - Volume 1*, IJCAI'87, pages 304–307, San Francisco, CA, USA, 1987. Morgan Kaufmann Publishers Inc.

[124] J. Ross Quinlan. *C4.5: programs for machine learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[125] Vibhor Rastogi, Nilesh Dalvi, and Minos Garofalakis. Large-scale collective entity matching. *Proc. VLDB Endow.*, 4(4):208–218, January 2011.

[126] Nalini K. Ratha and Ruud Bolle. *Automatic Fingerprint Recognition Systems.* SpringerVerlag, 2003.

[127] P. Ravikumar and W. Cohen. A hierarchical graphical model for record linkage. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 454–461, Arlington, Virginia, United States, 2004. AUAI Press.

[128] Shu Rong, Xing Niu, EvanWei Xiang, Haofen Wang, Qiang Yang, and Yong Yu. A machine learning approach for instance matching based on similarity metrics. In Philippe Cudr-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jrme Euzenat, Manfred Hauswirth, JosianeXavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *The Semantic Web ISWC 2012*, volume 7649 of *Lecture Notes in Computer Science*, pages 460–475. Springer Berlin Heidelberg, 2012.

[129] Bertrand Russell. On denoting. *Mind*, 14(56):pp. 479–493, 1905.

[130] R. C. Russell, 1918.

[131] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–278, New York, NY, USA, 2002. ACM.

[132] L. Sauermann, R. Cyganiak, and M. Vlkel. Cool uris for the semantic web. http://www.dfki.uni-kl.de/ sauermann/2006/11/cooluris/, August 2007. Revisioned version 1.1.

[133] W. Shen, X. Li, and A. Doan. Constraint-based entity matching. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 2*, pages 862–867. AAAI Press, 2005.

[134] P. Singla and P. Domingos. Entity resolution with markov logic. In *Data Mining, 2006. ICDM '06. Sixth International Conference on*, pages 572 –582, dec. 2006.

[135] Jennifer Sleeman and Tim Finin. A machine learning approach to linking foaf instances. In *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*, 2010.

[136] David Smiley and Eric Pugh. *Apache Solr 3 Enterprise Search Server.* Packt Publishing, 1st edition, November 2011.

[137] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Molecular Biology*, 147:195–197, 1981.

[138] H. Stoermer and P. Bouquet. A novel approach for entity linkage. In *Proceedings of the IEEE International Conference on Information Reuse and Integration, IRI 2009*, pages 151–156, 2009.

[139] Heiko Stoermer, Nataliya Rassadko, and Nachiket Vaidya. Feature-based entity matching: The fbem model, implementation, evaluation. In Barbara Pernici, editor, *Advanced Information Systems Engineering*, volume 6051 of *Lecture Notes in Computer Science*, pages 180–193. Springer Berlin / Heidelberg, 2010.

[140] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203 – 217, 2008. ¡ce:title¿World Wide Web Conference 2007Semantic Web Track¡/ce:title¿.

[141] Robert L. Taft. *Name Search Techniques.* Bureau of Systems Development, 1970.

[142] Jie Tang, Bang-Yong Liang, Juanzi Li, and Kehong Wang. Risk minimization based ontology mapping. In Chi-Hung Chi and Kwok-Yan Lam, editors, *Content Computing*, volume 3309 of *Lecture Notes in Computer Science*, pages 469–480. Springer Berlin Heidelberg, 2004.

[143] S. Tejada, C.A. Knoblock, and S. Minton. Learning object identification rules for information integration. *Information Systems*, 26:607–633(27), December 2001.

[144] Rattapoom Tuchinda, Craig A. Knoblock, and Pedro Szekely. Building mashups by demonstration. *ACM Transactions on the Web (TWEB)*, 5(3), July 2011. http://dx.doi.org/10.1145/1993053.1993058.

[145] Nicholas Unwin. The individuation of events. *Mind*, 105:315–330, 1996.

[146] Dirk van Dalen. *The Blackwell Guide to Philosophica Logic*, chapter Intuitionistic Logic, page 224257. Blackwell, Oxford, 2001.

[147] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data. In *The Semantic Web - ISWC 2009*, volume 5823 of *Lecture Notes in Computer Science*, pages 650–665. Springer Berlin / Heidelberg, 2009.

[148] Jiannan Wang, Tim Kraska, Michael J. Franklin, and Jianhua Feng. Crowder: crowdsourcing entity resolution. *Proc. VLDB Endow.*, 5(11):1483–1494, July 2012.

[149] Jiannan Wang, Guoliang Li, Jeffrey Xu Yu, and Jianhua Feng. Entity matching: how similar is similar. *Proc. VLDB Endow.*, 4(10):622–633, July 2011.

[150] J.R. Wang and S.E. Madnick. The inter-database instance identification problem in integrating autonomous systems. In *Data Engineering, 1989. Proceedings. Fifth International Conference on*, pages 46 –55, feb 1989.

[151] M.S. Waterman, T.F. Smith, and W.A. Beyer. Some biological sequence metrics. *Advances in Math*, 20:367–387, 1976.

[152] Xue wen Chen and Jong Cheol Jeong. Enhanced recursive feature elimination. In *Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on*, pages 429–435, 2007.

[153] Steven Euijong Whang, Omar Benjelloun, and Hector Garcia-Molina. Generic entity resolution with negative rules. *The VLDB Journal*, 18(6):1261–1277, 2009.

[154] Alfred North Whitehead and Bertrand Russel. *Principia Mathematica.* Cambridge, 1910.

[155] W. E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods (American Statistical Association)*, 1990.

[156] W.E. Winkler. ”improved decision rules in the felligi-sunter model of record linkage”. Technical Report Statistical Research Report Series RR93/12, US Bureau of the Census, Washington D.C., 1993.

[157] W.E. Winkler. Methods for record linkage and bayesian networks. Statistical Research Report RRS2002/05, US Bureau of the Census, Washington, D.C., 2002.

[158] Ningning Wu, John Talburt, Chris Heien, Nick Pippenger, Chia-Chu Chiang, Elizabeth Pierce, Ebony Gulley, and JaMia Moore. A method for entity identification in open source documents with partially redacted attributes. *J. Comput. Small Coll.*, 22(5):138–144, 2007.

[159] Chuan Xiao, Wei Wang, Xuemin Lin, and Jeffrey Xu Yu. Efficient similarity joins for near duplicate detection. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 131–140, New York, NY, USA, 2008. ACM.

[160] Huimin Zhao and Sudha Ram. Entity identification for heterogeneous database integration: a multiple classifier system approach and empirical evaluation. *Inf. Syst.*, 30(2):119–132, 2005.

[161] Huimin Zhao and Sudha Ram. Entity matching across heterogeneous data sources: An approach based on constrained cascade generalization. *Data Knowl. Eng.*, 66(3):368–381, September 2008.