Data Linking for the Semantic Web

Alfio Ferrara[i]
DICo, Università degli Studi di Milano, Italy

Andriy Nikolov
Knowledge Media Institute, The Open University, United Kingdom

François Scharffe
LIRMM, University of Montpellier, France

ABSTRACT

By specifying that published datasets must link to other existing datasets, the 4th linked data principle ensures a Web of data and not just a set of unconnected data islands. We propose in this paper the term data linking to name the problem of finding equivalent resources on the Web of linked data. In order to perform data linking, many techniques were developed, finding their roots in statistics, database, natural language processing and graph theory. We begin this paper by providing background information and terminological clarifications related to data linking.
We then provide a comprehensive survey over the various techniques available for data linking. We classify these techniques along the three criteria of granularity, type of evidence, and source of the evidence. Finally, we survey eleven recent tools performing data linking and we classify them according to the surveyed techniques.

Keywords: Semantic web; data linking; instance matching; linked data; record linkage; object identification; ontology matching; matching system

INTRODUCTION

In the Semantic Web and in the Web in general, a fundamental problem is the comparison and matching of data and the capability of resolving the multiplicity of data references to the same real-world objects, by defining correspondences among data in form of data links. The data linking task is becoming more and more important as the number of structured and semistructured data available on the Web is growing. The transformation of the Web from a "Web of documents" into a "Web of data", as well as the availability of large collections of sensor generated data ("internet of things"), is leading to a new generation of Web applications based on the integration of both data and services. At the same time, new data are published every day out of user generated contents and public Web sites.

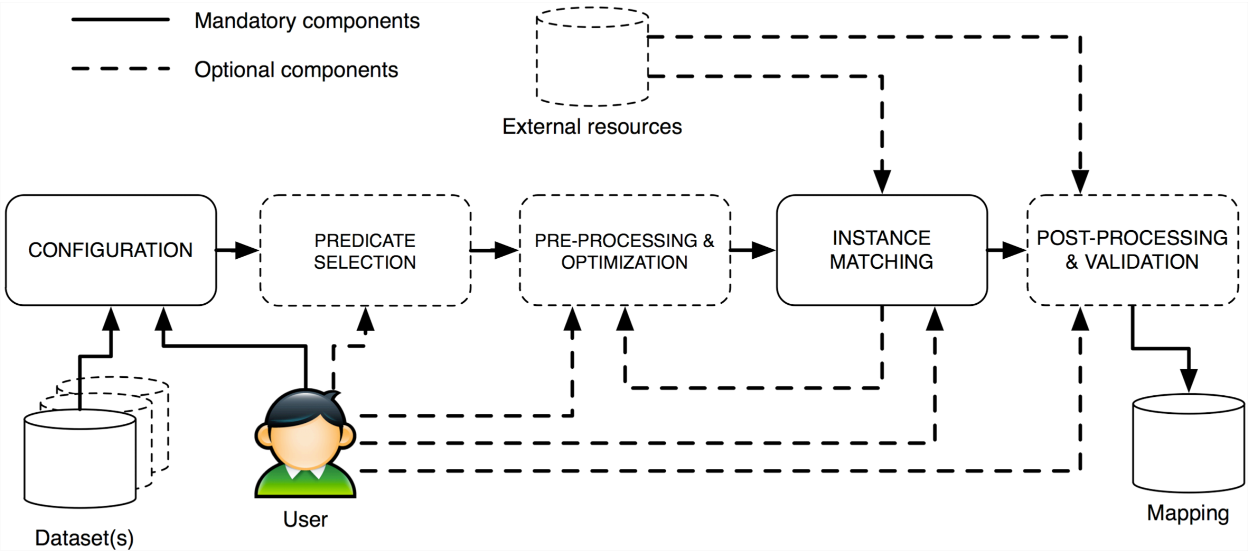
In general terms, *data linking* is the task of determining whether two object descriptions can be linked one to the other to represent the fact that they refer to the same real-world object in a given domain or the fact that some kind of relation holds between them[ii]. Quite often, this task is performed on the basis of the evaluation of the degree of similarity among different data instances describing real-world objects across heterogeneous data sources, under the assumption that the higher is the similarity between two data descriptions, the higher is the probability that the two descriptions actually refer to the same object. From an operative point of view, data linking includes also the task of defining methods, techniques and (semi-)automated tools for performing the similarity evaluation task. We call this specific subtask *instance matching*.

In this context, one of the most important initiatives in the Semantic Web field is the Linked Open Data project, which promotes the idea of improving interoperability and aggregation among large data collections already available on the Web (Berners-Lee et al., 2008). The example of Linked Data shows how the data linking task is crucial on the Web nowadays. Fortunately, there is a lot of work done in other fields that can be used to provide solutions, methods, and techniques to address data linking on the Semantic Web. Moreover, there are important works describing research fields that are very close to data linking (Euzenat & Shvaiko, 2007; Koudas et al., 2006; Bleiholder & Naumann, 2009). However, there are specific features of semantic data which require specific solutions both in terms of new techniques and in terms of original combinations of techniques that have been originally proposed in different fields. For example, on one side, data linking requires to deal with the semantic complexity which is typical of ontology matching, but, on the other side, the large amount of data available on the "Web of data" requires to deal with scalability issues that are typical of record linkage. This situation leads to the development of new approaches, addressing problems that are typical of the data linking field on the Semantic Web. In this paper, we provide a general definition of this field, in order to underline problems and to describe solutions. In particular, in the next section, we will better define the data linking problem, by discussing also the terminology used in the field. After that, we present the main families of techniques proposed for the most relevant subtask of data linking, that is instance matching, then we survey the most relevant tools in the field by comparing them. Finally, we discuss the main open problems and directions of work in the field.

PROBLEM FORMULATION

Data linking can be formalized as an operation which takes two collections of data as input and produces a collection of mappings between entities of the two collections as output. Mappings denote binary relations between entities corresponding semantically one to another. The data linking task is articulated in steps as shown in Figure 1.

Figure 1. The data linking tasks



The input of the process is given by one or more datasets. Each dataset is a collection of data representing object descriptions to be linked. The output of the process is a *mapping set* that is a collection of binary relations (usually referred as *mappings* or *links*) between the object descriptions in the input dataset(s)[iii]. The data linking task also involves a user, who has the responsibility of configuring the process and, optionally, interacting with the predicate selection, pre-processing, matching and post-processing steps in case of a semi-automatic procedure. Another optional component is provided by external resources such as lexical databases and thesauri, reasoning systems or pre-defined mappings that can be used both in the matching and in the post-processing steps as a support for the main process. A typical external resource used for the comparison is a schema level mapping that is used to determine which data must be compared. The main steps of the process are more extensively described in the following:

Configuration. The configuration step has the goal of setting up the parameters used during the instance matching step in order to compare object descriptions. In particular, it is very common to evaluate similarity between object descriptions as a value in the range [0,1] and to set a threshold that denotes the minimum value of similarity needed in order to consider a pair of object descriptions as similar one to the other. Another parameter in some systems is the choice of similarity metrics combination that has to be used in the matching step as well as the list of external resources that may be needed for similarity evaluation.

Predicate selection (optional). The general goal of the whole data linking task is to retrieve a set of links between two collections of data. The term “links” is generic and, in fact, the choice of the predicates used to represent the meaning of the linking relation is an important step of the data linking task configuration. Usually, the default option is to intend data linking as the task of retrieving links between data that refer to the same real objects, i.e., *identity* links. However, data linking techniques and tools can be used (and actually are often used) in order to retrieve more generic similarity relations among data, in order, for example, to exploit these links for approximate query answering. Since the goal of data linking depends on the assumptions and goals of users and it affects the nature of matching techniques that are enforced, predicate selection is part of the data linking configuration, even if some work exists about the problem of *a posteriori* mapping interpretation (see next section).

Pre-processing & optimization (optional). Pre-processing of data is an optional step that can be executed for two main purposes. The first is to transform the original representation of data according to a reference format used for the comparison. A second goal is to minimize the number of comparisons that have to be executed in order to produce the final mapping set. To this end several kinds of blocking techniques can be adopted to compare each object description only against those descriptions that have a high probability to be considered similar to it.

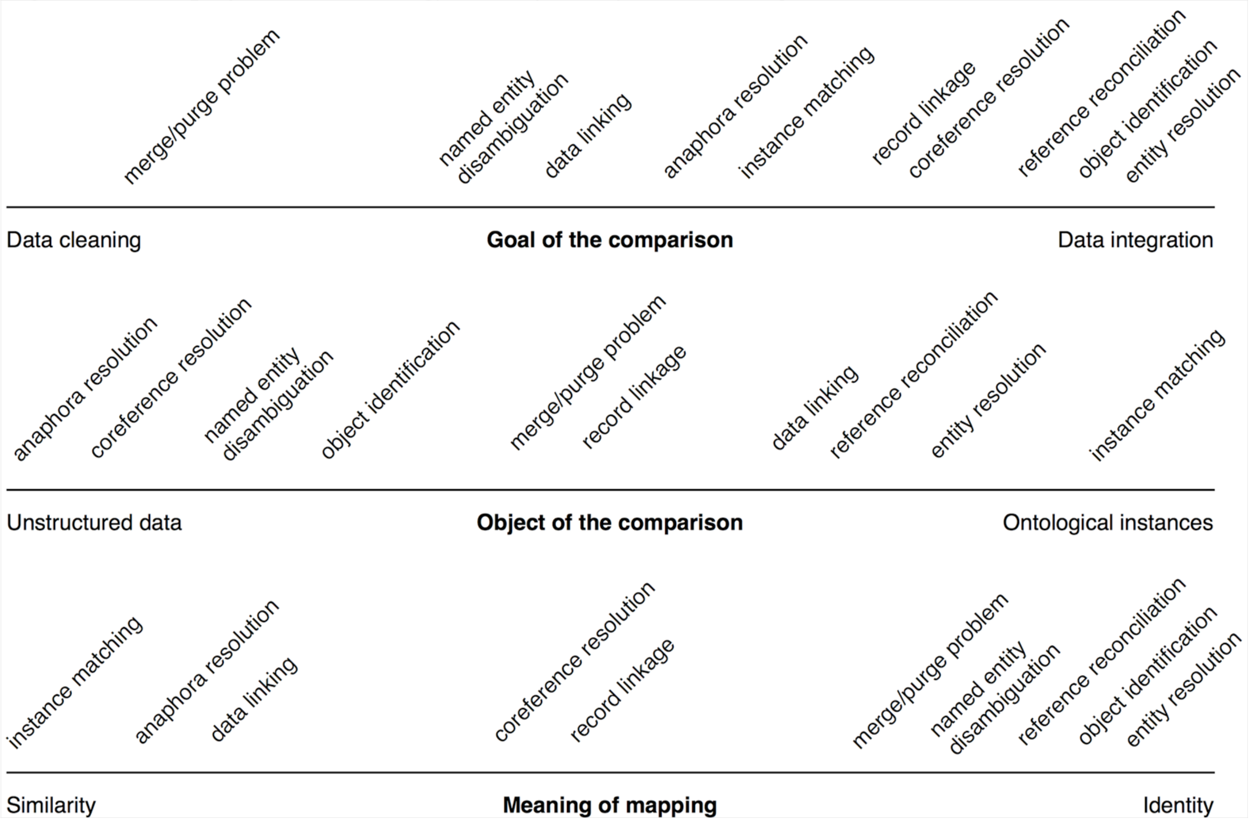
Matching. In the instance matching step, the object description comparison is executed according to the metrics chosen in the configuration step. In many cases, more than one type of matching techniques are combined, including for example string/value matching, learning-based matching, similarity propagation. If required, external resources are used in this step to optimize the matching with respect to a pre-defined mapping or to determine the similarity between property values according to existing lexical resources or ontologies (e.g., WordNet, SKOS, OpenCyC). In case of a semi-automatic process, the user interaction is needed in the matching step in order to select the correct mappings or to validate the system result.

Post-processing & validation (optional). The post-processing step is optional and has the goal of refining the matching result according to specific domain or application requirements. A typical post-processing goal is to validate the mappings with respect to a formal definition of consistency for the mapping set, that may include, for example, the fact that two object descriptions classified in disjoint classes cannot be mapped onto the same entity (Meilicke et al., 2008). Another kind of validation is executed with respect to the required mapping set cardinality. In some cases, a mapping set cannot include more than one matching counterpart for each object description. This requirement can be taken into account either in the instance matching step or in the post-processing step, by defining appropriate heuristics for the deletion of multiple mapping rules and the selection of the best ones (Castano et al., 2008).

Terminological hints and related problems

The problem of data linking is directly related to many similar problems, and this causes confusion about the appropriate terminology. Problems similar to data linking can be grouped in two main categories: the first category describes the problems related to the proper activity of *data linking* (Alexander et al., 2009), i.e. connecting together different data provided by heterogeneous data sources. In this field, we often talk about *entity matching*, *coreference resolution* or *entity resolution* to denote the activities of evaluating the degree of similarity between pairs of entities with the goal of finding a common reference to a single real-world object (Cudré-Mauroux et al., 2009; Köpcke et al., 2010; Köpcke & Rahm, 2010). On the other side, a second category of problems is related to the comparison of data records especially for data cleaning and duplicate detection purposes. In this field, we often talk about *duplicate identification*, *record linkage* or *merge/purge problem*. For both the categories of problems many solutions have been proposed. Recent surveys and contributions on this field include for example Koudas et al. (2006); Bleiholder & Naumann (2009); Batini & Scannapieco (2006). The main differences between these two categories of problems and their related solutions can be described according to three main dimensions of analysis: i) the *goal* of the comparison; ii) the *object* of the comparison, and iii) the *meaning* of the resulting mappings, as graphically shown in Figure 2.

Figure 2. A map of the terminology in the field of data linking



Goal of the comparison. Some of the problems mentioned above have the goal for example of detecting duplicate records in order to clean a collection of data by capturing errors. This is the case for example of the merge/purge problem in the context of data cleaning. Another goal is data integration. In this case, the final purpose is to provide a unique description for each real-world object mentioned in different data sources, by exploiting object identification and entity reconciliation/resolution techniques. Between these two options, a third goal is the linkage of data. In this case, we admit the presence of different and heterogeneous descriptions of real-world objects, but we want to create links and relations among these descriptions in order to make explicit their correspondence. To this end, several techniques may be used referred with terms such as “coreference resolution”, “record linkage”, and “instance matching”.

Object of the comparison. By object of the comparison we mean the kind of data that are taken into account for the data linking task. A general but useful distinction concerns techniques and solutions mainly focused on unstructured data, such as plain text or Web pages, on one side, and those mainly focused on richly structured ontology instances on the other side. Between these two sides of the comparison, it is very important to recall several methods and techniques that have been aimed at other types of structured data (relational or XML database instances). On the side of unstructured data, we collocate approaches mainly derived from the natural language processing field, such as anaphora resolution, named entity disambiguation, and coreference resolution (this last term is used also in the context of ontology instances). In the database community, the terms mainly used are record linkage and merge/purge. Terms like entity resolution, data linking, reference resolution are more typical of the Semantic Web and the ontology fields. The main difference between these kinds of solutions is that the assumptions that can be made on unstructured and relational data are different from those that are valid in case of ontological data. For example, in case of relational data we usually do not deal with multi-valued attributes and class hierarchy, and the structure of data is explicit. On the contrary, the structure of ontological data is implicit and must be inferred from the ontology. Moreover, the structure of ontological instances is often not rigid and can provide multi-valued or structured attributes. Finally, when dealing with relational data, the attribute values of records often consist in atomic data, and references to other records through foreign keys are not so frequent. On the contrary, in linked data it is very common to provide attribute values in form of references to other objects inside or across datasets.

Meaning of the mappings. Finally, about the meaning of mappings produced as a result of the comparison, we can distinguish between two main approaches. On one side, mappings are interpreted as “same-as” relations, where the two mapped object descriptions are intended to denote the same real-world object. On the other side, a mapping may be intended just as a generic relation of similarity between object descriptions. In those cases where the terminology mainly refers to the idea of connecting data more than finding a common reference to a real-world object, we have classified the terms in the category of similarity, such as in case of the terms instance matching, data linking, and anaphora resolution. Instead, terms like object identification, (named) entity resolution, reference reconciliation are used to stress the fact that the meaning of mappings found by the comparison task is to denote an identity relation between objects described by data. Other terms like coreference resolution and record linkage are used in both senses according to different approaches in the literature. More details about the interpretation of mappings meaning is given below.

Other relevant terminology and fields.

The problem of data linking and its associated techniques of instance matching and object description comparison is very pervasive also in fields other than the Semantic Web and for purposes other than the linking of data. For example, the expression *reference reconciliation* has been used with reference to ontological data in Dong et al. (2005) and Säis et al. (2007). The problem of comparing different object descriptions has been addressed also in the natural language processing research community, where linguistic descriptions of objects like names of sentences are compared for disambiguation purposes. In this case we talk about *anaphora resolution* or *named entity disambiguation* (Elmagarmid et al., 2007). In the database and data mining fields, the problem is also addressed as *object identification*.

Mappings and their meaning

The data linking task produces a collections of mappings that are used to define links between object descriptions in different data sources. In this context, a relevant issue is to determine the meaning of these mappings. Usually, the mappings are interpreted as binary relations between two object descriptions that could be considered *identical* in the sense that they refer to the same real-world object. The notion of identity in itself, however, is ambiguous and is a subject of philosophical discussions. In the literature about ontologies and the Semantic Web, identity has been described as the problem of distinguishing a specific instance of a certain class from other instances (Guarino & Welty, 2002). Thus, the identity criterion is linked to the class to which an instance belongs and depends on the assumptions made by the ontology designer. In this way, identity mappings between instances should, in principle, follow the identity criterion associated with the corresponding classes. However, instance matching usually deals with datasets originating from different sources, and the assumptions of their creators may be potentially different. Moreover, the design practices in the Semantic Web are always not explicitly declared and do not follow a standard approach. On the Web of Data, identity is commonly expressed using the standard *owl:sameAs* property. In OWL, two individuals connected by the *owl:sameAs* property are considered identical in the sense that the subject and object of this statement share all their properties. Such interpretation of identity, however, appears too strong in many cases and it is not always compatible with the evaluation of mappings, which is based on similarity. In order to understand the practices of the usage of *owl:sameAs* links by existing repositories, Halpin et al. (2010) distinguished five weaker varieties of identity beyond the canonical one. The idea of providing more than a single notion of identity is also adopted by the UMBEL and the SKOS vocabularies[iv], which provide weaker relations of correspondence such as *umbel:isLike* or the family of “Match” properties of SKOS.

More generally, the problem of having a sound and well founded notion of identity is often ignored in the data linking process. As a consequence, the interpretation of mapping can shift among three very different notions of identity:

1. *Ontological identity:* the idea behind ontological identity is that the real-world and existing object denoted by two identical object descriptions is the same. In other terms, we ask whether two “identical” object descriptions denote the same real-world object. For example, the expressions “Bard of Avon” and the name “William Shakespeare” denote the same real-world person.
2. *Logical identity:* what we call “logical identity” is the idea that two descriptions are identical when they can be substituted one to the other in a logical expression without changing the meaning of the expression. This kind of identity depends on the context where the expression is used and the fact that an object description is used to denote the reality of just to mention the term. For example, we can say “The Bard of Avon wrote A Midsummer Night’s Dream” instead of “William Shakespeare wrote A Midsummer Night’s Dream”. But, we cannot substitute the expression “The Bard of Avon” with “William Shakespeare” in a sentence like “William Shakespeare is often called the Bard of Avon”.
3. *Formal identity:* in case of formal identity, we refer to the fact that identity is superimposed to the data. This happens for example when some organization sets a unique identifier for something with the goal of providing a standard tool for the identification of the object, such as for example the ISBN code for publications.

The main problem with the notion of identity in the field of data linking is that the process of matching is always based on the notion of similarity and that the relation between similarity and identity is not clear. This can be shown by the following example: suppose we need to compare two object descriptions referred to different editions of the same book. Now, two descriptions are probably similar, but they can be considered identical only if we are interested in the book, seen as a work and not as the edition. But if we look for the manifestation of the book (i.e., the edition) or even for a physical volume in a library, these two descriptions have to be considered different. Thus, identity is not “internal” to the data and it is not directly dependent on the relation of similarity, but depends on the reference to something else than data (e.g., in this case the kind of object we are interested in). However, despite the fact that the identity problem is widely recognized and discussed in the research community and that ontologies defining weaker degrees of identity exist (e.g., SKOS or the ontology proposed in Halpin et al. (2010)), *owl:sameAs* remains the most popular representation of identity links between Semantic Web repositories. In literature, there is a variety of predicates proposed for denoting the general notion of "correspondence"[v]. Of course, having a canonical set of properties would be useful and might help in clarifying the task of mapping interpretation, but it is also an interesting research direction to work on the definition of semi-automatic approaches to mapping interpretations based on the nature of matching operations that are enforced during the linking task. At the moment, existing matching systems do not explicitly operate with different interpretations of matching results: instead the choice of interpretation is left to the user of the system.

INSTANCE MATCHING TECHNIQUES

As was discussed in the previous section, the problem of data linking is closely related both to the problem of database record linkage and the problem of ontology schema matching. Data linking systems processing semantic data utilize many techniques originating from both these communities. Thus, in order to determine suitable classification criteria for data linking techniques, we considered both the techniques themselves and existing classification schemas proposed in the literature on database record linkage (Elmagarmid et al., 2007; Winkler, 2006; Köpcke & Rahm, 2010) and schema matching (Rahm & Bernstein, 2001; Euzenat & Shvaiko, 2007). Based on this analysis, we have chosen the following dimensions for classifying linked data resolution techniques and approaches.

The first dimension we included in our classification schema is *Granularity*, which is also commonly used in the literature on adjacent topics. In the record linkage research there is a distinction between *field matching* and *record matching* techniques, which focus on two different stages of the linkage process (Elmagarmid et al., 2007). The former concentrate on identifying equivalent values for the same field in two records. This requires resolving the cases of different representation for the same data, such as misspellings, format variations, and synonymy (e.g., "Smith, John" and "J. Smith" or "Yangtze River" vs. "Chang Jiang"). The latter focus on deciding whether two records (possibly containing several fields) refer to the same real-world object. A similar distinction exists in the schema matching research community: e.g., as discussed by Rahm & Bernstein (2001) and Euzenat & Shvaiko (2007), matching algorithms are classified into *element-level* and *structure-level* ones. When analyzing existing data linking techniques with respect to the granularity criterion, we identified three main categories:

1. *Value matching.* Similarly to the field matching task in record linkage, this step focuses on identifying equivalence between property values of instances. In the simplest cases (e.g., when equivalence of two individuals is decided based on equivalence of their labels), no further steps are performed.
2. *Individual matching.* The goal of this stage is to decide whether two individuals represent the same real-world object or not. Individual matching techniques compare two individuals and utilize the results of the value matching stage applied to properties of these individuals. At this stage, two individuals are considered in separation from all other individuals in two datasets.
3. *Dataset matching.* This step takes into account all individuals in two datasets and tries to construct an optimal alignment between these whole sets of individuals. The techniques

handling this stage take as their input the results of the individual matching and further refine them. At this level, mutual impact of pairwise individual matching decisions can be taken into account: e.g., if we know that both datasets do not contain duplicates, then one individual cannot be mapped to two individuals from the other dataset.

As the second classification criterion, we use *Type of Evidence* used by the matching method. Based on this criterion, we distinguish between two categories of methods:

- *Data-level.* These methods rely on information defined at the level of individuals and their property values. In case of ontologies based on description logic, this information corresponds to ontological A-Box.
- *Knowledge-level.* These methods utilize knowledge defined by the ontological schema (e.g., subsumption relations, property restrictions) as well as in external sources. External sources can include, for example, third-party ontologies as well as linguistic resources which specify the meaning of terms based on their relations with other terms (such as WordNet (Miller, 1995).

Finally, we distinguished algorithms based on the *Source of Evidence*:

- *Internal.* These techniques only utilize information contained in the datasets being matched.
- *External.* These techniques exploit information contained in external sources.

To select the techniques for this section, we reviewed different relevant tools and algorithms described in the literature, which included the instance matching task in their process. We use the following criteria for considering a tool or an algorithm as a relevant one:

- It has to operate on semantic data expressed in RDF. Thus, various record linkage tools, which assume relational data representation are not included and the methods used in these tools are not mentioned. However, we consider also techniques developed for other types of data, if those techniques were later adopted by the tools processing semantic data.
- It has to include the instance matching task in its workflow. We primarily focused on the systems dedicated to data linking (see next section). However, we also included the techniques used by tools, which performed instance matching as an auxiliary task. In particular, these include generic ontology matching systems which match instance data in addition to schemas (e.g., RiMOM (Li et al., 2009) or ASMOV (Jean-Mary et al., 2009)), semantic search tools (e.g., PowerAqua (Lopez et al., 2009)), and identity management servers (e.g., OKKAM (Bouquet et al., 2008)).

With respect to the data linking workflow shown in Figure 1, most systems implement these techniques as a part of the *instance matching* stage. However, some of the techniques (primarily, dataset matching ones) are sometimes applied during the *post-processing & validation* step.

Table 1 summarizes the main categories of techniques and knowledge resources used by Semantic Web tools. In the rest of the section, we discuss each category of techniques in more detail.

Table 1. Categories of techniques applied in the Semantic Web domain

	Data-level		**Knowledge-level**	
	Internal	**External**	**Internal**	**External**
Value Matching	string similarity	keyword search services	-	linguistic resources formal domain models multi-lingual dictionaries
Individual Matching	attribute-based similarity	external mapping sets	schema restrictions	schema mappings external ontologies
Dataset Matching	belief propagation	external mapping sets	belief propagation (enhanced with ontological reasoning) linear programming	schema mappings

Value matching

Sometimes two individual descriptions contain the same property value expressed in different ways, e.g., because of different formatting conventions of two repositories or the use of synonyms. Resolving these discrepancies and determining that two values are equivalent constitutes the value matching task. As their output, value matching techniques often produce a score denoting the degree of similarity between two values, e.g., that “J. Smith” and “Smith, John” are likely to be two representations of the same name. Although equivalence of values does not necessarily imply equivalence of entities (e.g., there can be many people with the name “John Smith”), value matching techniques serve as building blocks in the instance matching process: their output is aggregated and serves as evidence for making decisions about equivalence of individuals.

Data-level techniques. Data-level value matching techniques primarily involve various *string similarity* measures widely used in both record linkage and schema matching. These measures often serve as basic components for more sophisticated methods operating at individual matching and dataset matching levels. External data-level methods are less frequent and include, in particular, standard *keyword search services* such as Google.

Internal approaches. To perform value matching, data linking systems widely adopt methods developed for database record linkage, in particular, various string similarity measures. There are several survey papers reviewing these measures (e.g., (Cohen et al., 2003), (Elmagarmid et al., 2007), (Winkler, 2006)) and we refer the reader to these papers for the detailed description of different similarity functions. Given that there is no single best string similarity measure for all domains, data linking tools usually allow selection among several measures. In particular, existing tools use the following similarity functions:

- *String equality.* The basic similarity function, which returns 1 if two values are equal and 0 otherwise. It has the highest precision, but does not accept slight variations caused by typos or minor formatting differences. Nevertheless, it is used, in particular, by the popular Silk data linking tool as one of the options (Volz, Bizer, Gaedke, & Kobilarov, 2009).

- *Edit distance (Levenshtein).* Character-based similarity function which measures the number of primitive change operations needed to transform one string value into another. Examples of systems employing this metrics are described by Volz, Bizer, Gaedke, & Kobilarov (2009), Li et al. (2009), and Lopez et al. (2009).
- *Jaro.* The measure designed to take into account common spelling deviations. It is also employed in the Silk system (Volz, Bizer, Gaedke, & Kobilarov, 2009).
- *Jaro-Winkler.* A modification of the Jaro measure adjusted to give a higher score to values which share a common prefix. It is commonly used to compare person names. Examples of usage are given by Udrea et al.(2007), Säis et al. (2008), Volz, Bizer, Gaedke, & Kobilarov (2009), Nikolov et al. (2008a), Ioannou et al. (2008).
- *Monge-Elkan.* This similarity function first involves dividing input strings into a set of substrings and then computing the maximal aggregated similarity between pairs of tokens. In the comparison study described by Cohen et al. (2003), this function achieved the best average performance. Among data linking systems, its use is reported by Volz, Bizer, Gaedke, & Kobilarov (2009) and Nikolov et al. (2008a).
- *Soundex.* This is a phonetic similarity function, which tries to determine string values which are pronounced similarly despite being different at the character level. The approach proposed by Ioannou et al. (2008) uses Soundex similarity as part of their algorithm.
- *Token set similarity.* This function involves tokenizing the input strings and then comparing resulting sets of tokens using a common set similarity function (such as Jaccard score or overlap distance). It is employed by Volz, Bizer, Gaedke, & Kobilarov (2009) and Jean-Mary et al. (2009).
- *I-Sub.* This function proposed by Stoilos et al. (2005) specifically for the ontology matching task calculates similarity between two strings as a function of both their commonalities and their differences. For the purpose of instance matching, this technique is employed in ObjectCoref (Hu et al., 2011).

In order to optimize the use of value matching techniques and reduce the number of comparisons needed to match two datasets, the LIMES framework (Ngomo & Auer, 2011) implements an approach which allows estimating the similarity or a pair of values based on the known similarity between each of two values and an exemplar point in a metric space. This technique uses the properties of metric spaces such as symmetry ($s(x, y) = s(y, x)$) and triangle inequality ($s(x, z) \leq s(x, y) + s(y, z)$), so it can be used with the similarity functions which satisfy these properties: e.g., Levenshtein, but not Jaro-Winkler.

We can distinguish two common usage patterns of string similarity measures depending on the purpose of the system:

Semi-automated tools, which must be pre-configured by the user often include a wide range of similarity functions. The user can select suitable similarity functions depending on the task at hand. This particularly applies to data linking frameworks designed to deal with RDF data such as Silk (Volz, Bizer, Gaedke, & Kobilarov, 2009) or KnoFuss (Nikolov et al., 2008a).

The tools that implement an automated workflow usually employ a single function or an aggregation of several measures applicable to a wide range of domains. This approach is usually implemented in ontology matching tools which also deal with instance matching (e.g., RiMOM (Li et al., 2009), ASMOV (Jean-Mary et al., 2009), or ILIADS (Udrea et al., 2007)) and semantic search tools (e.g., PowerAqua (Lopez et al., 2009)).

Existing tools usually do not implement the similarity functions but reuse public API libraries containing these implementations, such as Simmetrics[vi] or SecondString[vii].

External approaches. Standard keyword-based Web search engines provide an additional source of evidence available to matching tools. Normalized Google distance (Cilibrasi & Vitanyi, 2007) uses the sets of hits returned by Google for two labels being compared as an indication of semantic similarity between these terms. The distance is defined as:

$$NGD(x,y) = \frac{\max(\log(f(x)),\log(f(y))) - \log(f(x,y))}{\log(M) - \min(\log(f(x)),\log(f(y)))}$$

where:

- *f(t)* is the number of Google hits for the search term *t*
- *f(x,y)* is the number of Google hits for the tuple of search terms *x* and *y*,
- *M* is a total number of pages indexed by Google.

This distance measure is adopted for the generic ontology matching task by Gligorov et al. (2007). One disadvantage of this measure is the time cost of the Web search operations, which complicates its use for data linking tasks potentially involving large number of value comparisons.

Knowledge-level techniques. Knowledge-level techniques try to discover similarity between property values based on some semantic relations between them. In order to obtain this information, the tools involve external knowledge sources. These sources can be classified into two main categories: *Linguistic resources* and *Formal domain models*.

Linguistic resources provide relations between words used in the property values. WordNet (Miller, 1995) is the most commonly used thesaurus which contains such relations as synonymy and hyponymy, which are used directly for value matching. In particular, it is employed by Castano et al. (2006), Gracia & Mena (2008), Scharffe et al. (2009), Lopez et al. (2009). Similarity between terms is computed using distance measures based on the length of path between two entities in the graph. For example, the term affinity function which is used in HMatch (Castano et al., 2006) is defined as

$$A(t,t') = \begin{cases} \max_{i=1..k} W_{t \to_i^i t'} & if\,(k \geq 1) \\ 0 & otherwise \end{cases}$$

where k is the number of paths between t and t' in the graph, $t \to_i^i t'$ is the i-th path of length n, $Wt \to_i^i t' = W_{1tr} \cdot W_{2tr} \cdot \ldots \cdot W_{ntr}$ is the aggregated weight of the i-th path, which is equal to the product of weights associated with each edge in the path. Weights of edges are set depending on the type of the corresponding relation (hyponymy, synonymy, and so on).

In a specific case where the datasets contain field values expressed in different languages, multi-lingual dictionaries are used to pre-process values and translate them into a default language (e.g., English). Translation procedures are applied, for example, by the OKKAM server (Bouquet et al., 2008).

More specialized relations between terms are represented in publicly available ontologies. High-level vocabularies such as SUMO[viii] and OpenCYC[ix] contain terms from a wide range of domains; others, like SKOS[x], provide a vocabulary for relating other domain vocabularies. In particular, taxonomic distance based on SKOS is used in RDF-AI (Scharffe et al., 2009). The approaches based on formal domain models, however, are less commonly used for the instance matching task than in the schema matching community. This is due to the fact that these models mainly cover common terms, which correspond to classes in the ontologies, rather than entity names, which denote labels of individuals.

Summary. The choice of value matching techniques depends the domain and type of data values. Performance of different string similarity functions is influenced by the formatting conventions of labels and acceptable degrees of variance (e.g., changing order of tokens and abbreviations). For example, Jaro-Winkler similarity is better suited for person names, while Levenshtein is more appropriate for cases where variance is not expected, and only typos must be recognized. Thus, two approaches are possible: choosing appropriate functions for specific tasks or employing an ensemble of similarity functions. The use of external resources can substantially improve the performance, but the scope of tasks in which these resources is limited: commonly used external resources mainly cover common terms rather than entity names.

Individual matching

Establishing links between individuals which refer to the same real-world objects constitutes the main goal of the matching process. Individuals are described using their properties as well as relations to other individuals in the dataset. Individual matching techniques decide whether two individuals taken from different datasets should be linked or not using descriptions of these individuals as evidence.

Data-level techniques. Data-level individual matching normally involves aggregating the results of value matching methods over the values of properties belonging to two individuals being compared. Different *attribute-based similarity functions* have been proposed to perform such aggregation. In addition, existing *external mapping sets* are utilized to infer mappings between individuals in the context of Linked Data cloud.

Internal approaches. The classical approach to individual matching includes aggregating the results produced by value matching techniques. This model for solving the record linkage problem was proposed in a seminal paper by Fellegi & Sunter (Fellegi & Sunter, 1969). The model assumes that there exist two lists of records (A and B) describing some real-world objects. The task of record linkage is to classify each pair of records from the set $A \times B$ into two sets: M (set of matched pairs) and U (set of non-matched pairs). Given that each real-world object a or b is described using records $\alpha(a)$ and $\beta(b)$, the classification decision is based on a comparison of two records expressed as a vector function $\gamma(\alpha(a), \beta(b)) = \gamma(a,b)$ (distance function). The authors propose a probabilistic model introducing conditional probabilities

$$m(\gamma) = P\left(\gamma\left[\alpha(a),\beta(b)\right]\middle|(a,b) \in M\right) = \sum_{(a,b)\in M} P\left(\gamma\left[\alpha(a),\beta(b)\right]\right)\cdot P\left((a,b)\middle|M\right)$$

and $u(\gamma)$ (that is similar for $(a,b) \in U$).

The obtained value $m(\gamma)/u(\gamma)$ is used to make a decision about the equivalence of two entities. Possible decisions are denoted as A_1 (positive link), A_2 (possible (uncertain) link), and A_3 (positive non-link). The decision is chosen by comparing the value $m(\gamma)/u(\gamma)$ with thresholds $T\mu$, such that if $m(\gamma)/u(\gamma) > T\mu$ then $P(A_1|U) < \mu$, and $T\lambda$, such as if $m(\gamma)/u(\gamma) < T\lambda$ then $P(A_3|M) < \lambda$, where μ and λ are desired error levels. The challenges in this classical model include calculating $m(\gamma)$ and $u(\gamma)$, threshold values $T\mu$ and $T\lambda$, and the form of the comparison function γ. In the original method proposed by Fellegi & Sunter (1969), components of the vector $\gamma = (\gamma_1, \gamma_2,\ldots,\gamma_K)$ are the results of pairwise field values comparison produced by value matching techniques.

This model serves as the base for the majority of algorithms developed in the record linkage domain (see Elmagarmid et al. (2007) for a survey) and was re-used in several tools developed in the Semantic Web community. The aggregation methods adopted by these tools include the following:

- *Weighted average.* $\gamma = \frac{\sum_{i=1}^{K} w_i\gamma_i}{K}$, where w_i denotes the weight of the ith element of the attribute comparison vector.
- *Picking maximal (minimal) value.* $\gamma = \max_{i=1}^{K}(\gamma_i)$
- *Euclidean distance.* $\gamma = \sqrt{\sum_{i=1}^{K}\gamma_i^2}$
- *Weighted product.* $\gamma = \prod_{i=1}^{K} w_i\gamma_i$
- *Group linkage.* $\gamma = \frac{\sum_{i=1}^{K} s_i}{|Q| + |C| - |M|}$, where $s_i = \{\gamma_i,\ if\ \gamma_i > t,\ 0\ otherwise\}$, $\{Q, C\}$ are the sets of predicated of two individuals, and M is the set of elements for which $\gamma_i > t$
- *Log-based tempered geometric mean.* $\gamma = \exp\frac{\sum_{i=1}^{K} w_i \log(\gamma_i + \varepsilon)}{\sum_{J=1}^{K} w_j} - \varepsilon$.

In particular, the Silk system (Volz, Bizer, & Gaedke, 2009) assumes a supervised matching scenario where the user selects an aggregation approach (weighted average, max(min), Euclidean distance, or weighted product) for her task. Similarly, the the ODDLinker system (Hassanzadeh et al., 2009) used to discover interlinks for the LinkedMDB repository represented RDF individuals as relational tuples and employed aggregated attribute-based similarity to decide on equivalence of individuals.

In contrast, the OKKAM Entity Naming Service (Bouquet et al., 2008) is targeted at the unsupervised matching scenario in an open environment and employs automatic selection of an aggregation function. In order to achieve high precision, more involved functions (such as group linkage or geometric mean) are utilized (Ioannou, Niederée, et al., 2010). The SERIMI system (Araujo et al., 2011) uses a two-step approach: first, it selects potentially matching candidates using string similarity applied to their labels, and then utilizes the specially designed CRDS similarity function over the complete set of instances properties to discriminate between several candidate matching pairs with similar labels. An alternative approach to the choice of a suitable attribute-based similarity function utilizes unsupervised learning using a genetic algorithm (Nikolov et al., 2011): the system uses assumptions about the matched datasets (e.g., the need for 1-to-1 mappings) and incorporates them into a fitness function, which is used by the genetic algorithm to propose an optimal parameters of the composite matching function: pairs of attributes to compare, specific value matching functions, weights, and the threshold.

External approaches. While internal data-level individual matching techniques are largely adopted from traditional record linkage research, external approaches often utilize the features specific for the Web environment, and, in particular, Semantic Web. Sets of pre-existing identity links between individuals declared in distributed Linked Data repositories constitute one such feature. Using the transitivity of the *owl:sameAs* property, these identity links can be aggregated to infer the equivalence between individuals connected via a chain of *owl:sameAs* links. This approach is directly exploited by the authors of *sameas.org* portal[xi], which aggregates equivalent Linked Data resources into "coreference bundles" using both explicit *owl:sameAs* links and inferred ones. The idMesh algorithm (Cudré-Mauroux et al., 2009) employs this method as its first step to construct the graphs connecting potentially equivalent RDF individuals, which it later refines using the graph analysis techniques. The ObjectCoref system (Hu et al., 2011) also uses existing sets of individuals linked via the *owl:sameAs* property to bootstrap learning of discriminative properties which are later used to infer new equivalence mappings.

Knowledge-level techniques. The main source of knowledge which can be utilized by individual matching algorithms are *ontological restrictions* defined by domain ontologies used to structure the data in the matched repositories. In cases where the repositories are structured using different schema ontologies, *ontology mappings* obtained from external sources can be exploited to establish correspondences between schema terms and align the structure of individual descriptions in both repositories.

Internal approaches. The main sources of information used by internal knowledge-level approaches are the domain ontologies which define the structure of individuals in the input repositories. Logical axioms defined in the ontologies can provide both "positive" evidence from which new equivalence mappings can be inferred as well as "negative" evidence which can be used to filter out spurious mappings. Because of this, methods based on the use of ontological constraints are often applied to refine an initial set of candidate equivalence mappings obtained using other methods (such as attribute-based similarity).

In particular, the following ontological constructs are valuable for the individual matching task and are often directly utilized by matching systems:

- *owl:FunctionalProperty* (*owl:InverseFunctionalProperty*). These properties essentially define the key attributes for individuals similarly to database keys. Equivalent values of inverse functional properties imply the equivalence of the subjects of properties. Functional properties, in a similar way, allow equivalence of objects to be inferred if they are connected to the same subject. Functionality provides a strong restriction which can be useful both for inferring the new *sameAs* links between instances and for detecting spurious existing links: if two individuals were initially considered as equivalent and connected by a *owl:sameAs* mapping but have two different values for the same functional property, this can indicate that the mapping was incorrect.

- *owl:maxCardinality.* Similarly to the functionality restriction, cardinality violation can also point out to an incorrect mapping (although, according to its definition, it does not directly imply equivalence of property objects).
- *owl:disjointWith.* As another type of negative evidence, disjointness between classes can be used to reject candidate mappings if they connect instances belonging to different classes.

Among the existing systems, Sindice (Tummarello et al., 2007) implements a method for instance matching using explicitly defined inverse functional properties as positive evidence to infer equivalence between instances. The L2R algorithm (Säis et al., 2007) utilizes all relevant ontological schema constraints mentioned above and uses them as both positive and negative evidence. ObjectCoref (Hu et al., 2011) also employs ontological restrictions together with explicit *owl:sameAs* links to create the initial "kernel" mappings. These "kernel" mappings are used as a training set to learn discriminative properties, which in turn help to infer more individual mappings.

While logical restrictions are valuable in the data linking task, it is difficult to exploit them if the datasets are structured using different schema ontologies: precise translation of ontological axioms is problematic for automatic ontology matching systems. ILIADS (Udrea et al., 2007) is an ontology matching system, which performs schema- and data-level matching in iterations. After each data-level matching iteration, restrictions defined in both ontologies being matched are used to check the validity of obtained mappings between instances. Then, at the schema-level iteration, these mappings are used as evidence to infer mappings between classes (by applying set similarity metrics over the sets of their instances).

External approaches. External knowledge sources are particularly valuable in the data linking tasks involving repositories structured using different ontologies. Such tasks are common in the Linked Data environment, and existing *schema mappings* and *schema restrictions* represent a commonly used knowledge resource.

In the approach described by Nikolov et al. (2010) schema mappings are inferred using existing *owl:sameAs* links between Linked Data repositories. In case where instances of two repositories are not linked to each other directly, but there exist connections to the same external repositories, such links are used as evidence for instance-based ontology matching.

For example, both instances *movie:music_contributor* from LinkedMDB[xii] and *dbpedia:Ennio_Morricone* from DBpedia[xiii] are connected to *music:artista16e47f5-aa54-47fe-87e4-bb8af91a9fdd* from MusicBrainz[xiv]. Aggregating sets of such composite mappings and using set similarity functions, the system can derive mappings between corresponding classes *movie:music_contributor* and *dbpedia:MusicalArtist*. Schema-level mappings obtained in this way present an additional input to the KnoFuss system (Nikolov et al., 2008a). They are used to determine subsets of two repositories which are likely to contain identical instances and to perform instance matching in the same way as in a single-ontology scenario.

An extension to the CIDER ontology matching system (Gracia & Mena, 2008) described in (Gracia et al., 2009) uses ontological context to disambiguate ontological entities (both classes and individuals), e.g., to disambiguate the country name "Turkey" from the name of the bird.

Ontological context consists of all neighboring ontological terms and is obtained by querying the Watson ontology search service[xv]. This ontological context is then used to cluster different senses of the same term available on the Semantic Web as a whole and to assign the ontological entities being compared into appropriate clusters. This approach, however, is primarily targeted at the schema level and has a limited applicability to instance matching. For instance, it is difficult to distinguish between two individuals belonging to the same class as there would be little difference in their ontological context.

Moreover, external knowledge bases available on the Web provide useful evidence for disambiguating data instances. For example, Wikipedia[xvi] provides a generic source of information used, in particular, for disambiguating data instances extracted from text (e.g., (Bryl et al., 2010)). For restricted domains examples of such knowledge bases include, e.g., UniProt[xvii] (for proteins) and KEGG GENES[xviii] (for genomics).

Summary. Individual matching techniques implement the core functionality of a data linking system: producing mappings between pairs of individuals. While these mappings can be further refined by dataset matching techniques, these techniques strongly rely on the output of individual matching and, in particular, confidence degrees associated with mappings. The choice of appropriate individual matching techniques must be motivated by several factors, among them:

- *Degree of semantic heterogeneity.* In a scenario involving datasets structured using the same ontology or two aligned ontologies, correspondences between properties must be utilized: it makes sense only comparing values of corresponding properties, as envisaged by the Fellegi-Sunter model. In the absence of such mappings, the system can either try to establish them ((Hu et al., 2011), (Nikolov et al., 2011)) or rely on "bag-of-words" techniques.
- *Availability of external reference sources.* The advantages of using trusted data sources (e.g., DBpedia, Freebase, or GeoNames) are the possibilities to involve additional information describing individuals, as well as to estimate the ambiguity of a particular description: e.g., whether a certain label (such as "Paris") denotes a unique entity ("Yangtze River") or is ambiguous ("Cambridge").
- *Availability of additional information exploitable by dataset matching algorithms.* The dataset matching stage can refine initial mappings produced by individual matching. If dataset matching is applicable, then recall of individual matching becomes particularly important: incorrect initial mappings can still be filtered out, while omissions are more difficult to recognize.

Dataset matching

Individual matching techniques analyze a pair of individuals taken from two datasets together with their attributes and produce a decision on whether these two individuals are matching or not. However, it is often the case that comparing individual descriptions alone is not sufficient and misses relevant additional information. In particular, this information includes:

- *A-priori knowledge about the matched datasets.* For example, knowing that one of the datasets does not contain duplicates gives an indication that out of several mappings linking to the same instance only one can be correct.

- *Information about relations between individuals in corresponding datasets.* For example, deciding that two publication references in two citation datasets refer to the same paper also implies that the references to their authors are also matching.

Such information can only be utilized by analyzing the whole set of potential mappings between two datasets. This is the focus of different dataset matching techniques, which are applied to the initial set of candidate mappings provided by the individual matching stage and refine it. In the domain of linked data, dataset matching techniques can be particularly valuable when matching against a trustable reference source (such as DBpedia or GeoNames). In this case, a-priori information about such a source can be exploited: for instance, that all instances in such sources are distinct, and that some relations are in fact functional even when not declared as such in the ontology (e.g., birth place of a person).

Data-level techniques. Data-level dataset matching techniques involve the analysis of graphs formed by both relations between individuals within repositories and identity mappings across repositories. Internal matching techniques perform this on the basis of two datasets being matched while external ones involve information from third-party sources, in particular, Linked Data repositories containing links to the input datasets. To reason about the data statements and individual mappings in these graphs, various *belief propagation frameworks* are used.

Internal approaches. Existing systems commonly employ various similarity propagation techniques. These techniques exploit the principle that the similarity between two nodes in the graph depends on the similarity between their adjacent nodes. A generic graph matching algorithm called similarity flooding (Melnik, 2002) is used both in schema- and data-level ontology matching (in particular, in the RiMOM system (Li et al., 2009)). Similarity flooding includes the following stages:

1. Transforming datasets into a directed graph in which pairs of entities (potential matches) correspond to nodes. Edges between two nodes exist in both directions if in both datasets there is a relation between the corresponding entities.
2. Assigning weights to the edges. Usually, $w_{ij} = 1/n$, where w_{ij} is the weight of the edge from the node *i* to the node *j* and *n* is the number of outgoing edges of the source node *i*.
3. Assigning initial similarity σ^0 to nodes. The value of σ^0 is usually taken from the output of the individual matching stage.
4. Computing σ^{i+1} using a weighted linear aggregation function. The default function is defined as

$$\sigma^{i+1}(x,x') = \sigma^0(x,x') + \sum_{e_p} \sigma^i(y,y') \times w(e_p),$$

   where $\sigma^i(x, x')$ is the similarity value for the node representing the mapping between entities x and x', $e_p = \ll y, y' >, p, < x, x' \gg \in G$ is an edge from the node $< y, y' >$ to the node $< x, x' >$ with the label p, and $w(e_p)$ is the weight of this edge. Resulting values σ^{i+1} are normalized after each iteration.
5. The procedure stops if no similarity value changes more than a particular threshold ε or after a pre-defined number of iterations.

In the algorithm proposed by Dong et al. (2005) another propagation algorithm is used where the graph includes not only individual matching nodes but value matching nodes as well. Nodes corresponding to pairs of individuals are connected by edges to nodes corresponding to pairs of their property values: e.g., a node representing a potential mapping between two publications *{a_1, a_2}* has edges from the node representing the similarity between titles {"Title_1", "Title_2"} and publication years {1978, 1979}, expressing the dependency of the individual similarity from value similarities. In this way, the propagation algorithm combines the value matching, individual matching, and dataset matching stages in a single workflow. At each iteration, the algorithm makes a decision about whether a pair of individuals should be merged. Each individual in a merged pair is assumed to contain all the properties of its counterpart, and the graph is updated accordingly. Impact factors which specify how a similarity between a pair of nodes influences similarities between pairs of their neighbors are chosen on the basis of the type of property corresponding to the edge and the class of instances included in the node. These parameters are set by human users.

In a similar way, the RDF-AI system (Scharffe et al., 2009) implements an algorithm which propagates similarities from value similarity nodes to resource similarity ones. This algorithm does not assume the use of the same schema ontology by two datasets. The procedure first selects pairs of matching property values. At each iteration, a pair of values with the highest similarity is selected as potentially matching and other candidate pairs containing the same values are discarded. Then, similarities are propagated from value matching nodes to individual matching ones. The aggregation function chosen by the authors calculates a new resource similarity as average of the neighboring value similarity nodes.

The approach presented by Ioannou et al. (2008) is based on the interpretation of similarities as Bayesian probabilities. Accordingly, Bayesian networks are used as the framework for belief propagation in graphs. The propagation is performed by the standard message passing algorithm described by Pearl (1988).

External approaches. External data-level dataset matching approaches aggregate mutual impact of individual matching decisions in a set of several repositories rather than only two at a time. This method is particularly relevant in the Linked Data environment which represents a network of distributed interconnected repositories. These interconnections are usually created by applying data linking to a pair of repositories. In the idMesh algorithm (Cudré-Mauroux et al., 2009), these sets of mappings between individuals are combined over the whole network of data repositories and refined by analyzing mutual impact of these mappings. In this way, idMesh can be considered a meta-level data linking algorithm. idMesh builds graphs based only on equivalence and non-equivalence relations between entities and uses factor-graph message passing to compute marginal probabilities. As mentioned in next section, ObjectCoref (Hu et al., 2011) uses a set of mappings between individuals collected from the whole network of distributed semantic repositories as a training set to learn discriminative data patterns.

Knowledge-level techniques. Dataset matching techniques often rely on relations between individuals as evidence. These relations can either be declared or assumed: e.g., having a hypothesis that all individuals within a repository are different from each other. To decide how a specific relation impacts a matching decision, data-level techniques utilize various statistic-based heuristics (e.g., number of incoming/outgoing properties in similarity flooding). One disadvantage of the purely data-level techniques is ignoring explicit definitions of relations provided by the domain ontologies. Knowledge-level methods aim at improving the dataset matching results by utilizing this information. These methods usually enrich the *belief propagation algorithms* operating at the data level. However, one of the proposed approaches also implements dataset matching as a standard *linear programming* problem.

Internal approaches. Internal knowledge-level dataset matching techniques utilize information defined in the domain ontologies which describe the structure of data in the repositories.

LN2R system (Säis et al., 2008) employs similarity propagation as a part of the "numerical" matching algorithm N2R, which is combined with the "logical" one (L2R). Unlike the algorithms described in next section, this procedure is schema-aware. It employs the aggregation function

$$\sigma^{i+1}(x,x') = \max(\sigma_f^{i+1}(x,x'),\sigma_{nf}^{i+1}(x,x')),$$

where $\sigma_f^{i+1}(x,x')$ is the aggregated similarity function over the edges representing functional properties and $\sigma_{nf}^{i+1}(x,x')$ corresponds to non-functional ones. Similar to Melnik (2002), the impact of non-functional attributes is reduced proportionally to the number of edges corresponding to the same property.

In the method described by Nikolov et al. (2008b) the results of value matching, ontological axioms, and relations between individuals are combined together using valuation networks (Shenoy, 1992). Valuation networks are graphs containing two kinds of nodes. Variable nodes correspond to the confidence degrees of individual matching decisions and data statements[xix]. Valuation nodes represent the rules that determine the valid combinations of states of the neighbor variable nodes and correspond to ontological axioms and weak relations. Confidence degrees are interpreted as Dempster-Shafer belief distributions (Shafer, 1976).

Unlike the majority of dataset matching approaches which are based on various algorithms for belief propagation on the graph, Noessner et al. (2010) implement the procedure for global optimization of the whole set of mappings. They use class subsumption relations in addition to property restrictions. Their method measures the impact of non-strict ontological relations relevant for a particular mapping between instances. For example, having two potential instance mappings *(a, b_1)* and *(a, b_2)* and the concept *C* defined in the ontology such that *$a, b_1 \in C$*, but *$b_2 \notin C$*, the first mapping is preferred because it better fits the semantics of the domain ontology. They define the A-Box similarity measure, which quantifies the degree of similarity between two ontological A-Boxes described in terms of the same ontological T-Box. This similarity metrics relies on the value of the overlap function *$overlap_T(A_1,A_2,M)$* between A-Boxes *A_1* and *A_2* induced by an instance alignment *M*. The weighted overlap function measures the aggregated "compatibility" of all pairwise mappings included in *M*.

$$overlap_T^w(A_1,A_2,M) = overlap_c + overlap_p + overlap_{\neg c} + overlap_{\neg p},$$

where

- $overlap_c = \sum_{(<a,b>)(C(a)\wedge C(b))} \sum \sigma(a,b)$, where *<a,b>* $\in M$ and $C \in T$ - the aggregated confidences *(σ(a,b))* of mappings connecting instances belonging to the same class,
- $overlap_p = \sum_{(<a,b>,<a',b'>)(P(a,a')\wedge P(b,b'))} \sum \frac{\sigma(a,b)+\sigma(a',b')}{2}$, where *<a,b>,<a',b'>* $\in M$ and $P \in T$ – impact of mappings between pairs of individuals connected by the same property in both repositories,
- $overlap_{\neg c} = \sum_{(<a,b>)(\neg C(a)\wedge \neg C(b))} \sum \sigma(a,b)$, and
- $overlap_{\neg p} = \sum_{(<a,b>,<a',b'>)(P(a,a')\wedge P(b,b'))} \sum \frac{\sigma(a,b)+\sigma(a',b')}{2}$ – impact of corresponding negated concept and property assertions.

At the next step, weighted A-Box similarity has to be maximized, in other words the value of $\arg\max_{M}(A_1, A_2, M)$ has to be determined. The authors propose two alternative approaches to achieve this. The first one involves transforming the problem into an integer linear programming problem (Schrijver, 1998) and solving it using standard methods. However, the disadvantage of these methods is their computational complexity, which makes it difficult to apply them to large-scale repositories. In order to reduce the computational cost, the authors included an alternative approach which involves applying a generic algorithm for inexact graph matching (Cour et al., 2006). This algorithm reduces execution time with some loss of precision.

External approaches. External knowledge sources can contribute with additional information of the same kind as utilized by internal techniques, namely ontological restrictions, subsumption and equivalence relations between schema terms, information about properties valuable for determining identity and so on. Relevant knowledge sources include, among others:

- Schema-level mappings obtained using third-party services. Such mappings can be used to translate ontological restrictions from the terminology of one ontology into another.
- Existing interlinked data repositories. These repositories can be used to infer schema-level mappings using instance-based ontology matching (as described by Nikolov et al. (2010)) and to mine common data patterns such as discriminative properties for matching (as described by Hu et al. (2011)).

Summary. As we can see, dataset matching techniques are primarily targeted at exploiting additional information about processed datasets. Although such additional information can be unavailable in the general case, the Semantic Web domain and linked data environment in particular possess specific features, which can be exploited by dataset matching methods. Such specific features include:

- Rich structure and additional expressivity provided by ontological schema definition languages (especially OWL) in comparison with relational database schemas. Advanced schema constructs (e.g., class hierarchy, disjointness relations) can be used to refine the mappings produced by individual matching and dataset matching methods.

- Availability of large volumes of publicly available data and existing mappings between them. In the Linked Data cloud, data linking systems can consider a whole network of repositories instead of only two as in classical record linkage and ontology matching scenarios.

DATA LINKING SYSTEMS

Data linking systems implement a number of the techniques identified in the previous section in order to interlink Web data described in RDF.

In the following analysis, we study 11 systems performing both automated and semi-automated data linking. While many ontology matching systems are also able match instance sets or use instance matching techniques in the ontology matching process (see Castano et al. (2006); Li et al. (2009); Gracia & Mena (2008); Jean-Mary et al. (2009); Noessner et al. (2010); Udrea et al. (2007)), we focus here on systems whose primary focus is to perform data linking. A few other systems dealing with Web data use data linking techniques while not focusing on linking data and are thus not included in this section (see Lopez et al. (2009); Bouquet et al. (2008); Tummarello et al. (2007)).

More generally, many ad-hoc matchers are designed when interlinking a specific dataset on the Web of data. Semi-automated data linking systems typically use configuration files that need to be adapted for each pair of datasets. Parameters such as matching techniques, properties to be compared and thresholds need to be entered by the user. This manual input is in most cases needed if a high link quality has to be reached. Automated matching with high quality links can be achieved if the domain of the tool is well defined, such as for LD Mapper (Raimond et al., 2008) for the music domain.

Tools are described below; then we specify which of the techniques we presented in the previous section have been implemented by each tool.

LN2R. LN2R (Säis et al., 2008) is an unsupervised instance matching system. It uses two approaches: one logical (L2R) and one numerical (N2R). L2R exploits the axioms of the ontologies describing instances. Functional and inverse functional properties, as well as disjoint classes axioms are considered. Matches and non-matches found by L2R are then used as input for the numerical similarity method N2R. N2R uses equations modeling dependencies between similarities of entities. This capture the intuition that if two pairs of instances are related in two datasets and two of these instances are similar across datasets, then the two other instances are likely to be similar as well. Similarity between attributes is computed using an external thesaurus or string similarity algorithms. Then an iterative algorithm computes the similarity of instances based on the attributes similarities and the dependencies equations.

ObjectCoref. ObjectCoref (Hu et al., 2011) aims at building a searchable repository of identity links between resources on the Web of data. The data linking system is based on a self-training network, a semi-supervised learning framework. The system thus needs a training set before being able to interlink two datasets. ObjectCoref uses ontological axioms to further discover equivalences between resources: existing *owl:sameAs* links, functional and inverse functional, as well as cardinality restrictions. A discriminating factor is then computed for pairs of properties belonging to matching resources in the training set. The system then performs an analysis of property-value pairs in order to ascertain which properties have a similar value for resources in the training set. In fact the system performs on the fly matching of properties used in the two datasets. The matching properties are then used to link the two datasets in an iterative process.

Okkam. Okkam (Ioannou, Niederée et al., 2010) proposes an architecture based on the use of distributed servers maintaining sets of equivalent resources. They are named Entity Name Servers (ENS). Each equivalent resource set is assigned a identifier. ENS store entity descriptions on the form of key/property values. New entities are added based on a matching algorithm constructing a similarity measure between the candidate resource and the ENS entity. The similarity measure uses a string matching algorithm between the key/property pairs. The similarity measure is then weighted according to the likelihood of the key to indicate a name for the entity. A small vocabulary of naming properties is thus maintained in the system. Finally similarities are aggregated by computing the sum of maximal similarities between the features of the two entities.

RKB-CRS. The co-reference resolution system of RKB (Glaser et al., 2008; Jaffri et al., 2008) consists of resource equivalence lists. These lists are constructed using ad-hoc Java code on the specific conference/university domain. Domain heuristics are provided such as co-authorship analysis. New code needs to be written for each dataset. It consists in selecting resources to be matched and performs string similarity matching on relevant attributes.

LD Mapper. LD Mapper (Raimond et al., 2008) is a Prolog based dataset interlinking tool. The tool is based on a similarity aggregation algorithm taking into account the similarity of neighbor resources. The tool requires little user configuration and has been tested on music related datasets. The current implementation works with the Music Ontology, but the algorithm can be used on datasets working with any ontology.

Silk. Silk (Volz, Bizer, Gaedke, & Kobilarov, 2009) is a framework and tool for interlinking datasets and maintaining the links. It consists of a tool and a link specification language: the Silk Link Specification Language. Before matching two datasets, the user specifies entities to link in a LSL file. The tool uses various string matching methods, but also numeric equality, date equality, taxonomical distance similarity, and sets similarity measure. All these similarity measures are parameterized by the user using a specific format. Preprocessing transformations can be specified by the user in order to improve the quality of the matching. Matching algorithms can be combined using a set of operators (MAX, MIN, AVG). Also, literals can be transformed before the comparison by specifying a transformation function, concatenating or splitting resources. Silk takes as input two datasets by specifying SPARQL endpoints. It is able to output *sameAs* triples or any other predicate between the matched entities. Silk was tested on diverse datasets available on its project Web page[xx].

LIMES. LIMES (Ngomo & Auer, 2011) is a semi-automatic interlinking tool that can be configured using a XML specification format. Its originality lies in the usage of the properties of metric spaces in order to optimize the use of matching techniques and to reduce the number of comparisons needed to match two datasets. The tool needs user input formatted using a link specification language. The tool is available online as a Web service[xxi] and a graphical user interface to a Web service is also available at[xxii].

KnoFuss. The KnoFuss architecture (Nikolov et al., 2008a) is designed for the fusion of heterogeneous knowledge sources. One particularity of KnoFuss is its ability to match datasets described under heterogeneous ontologies. The matching process is driven by an ontology specifying resources to be matched, and the adequate matching techniques to use for these resources. For each type of resource to be matched, an *application context* is defined, specifying a SPARQL query for this type of resource. A variety of string similarity algorithms are available. When datasets described by heterogeneous ontologies are used, it is possible to specify an ontology alignment in the alignment format[xxiii], thus allowing to use any matcher outputting this format. The tool is primarily meant to perform fusion of two input dataset. The input fusion ontologies thus also specify how should the resources be fused. A post-processing step is performed to verify and enforce the consistency of the new dataset resulting from the fusion operation with regards to the ontologies. The tool works with local copies of the datasets and is implemented in Java.

RDF-AI. RDF-AI (Scharffe et al., 2009) is an architecture and prototype implementation for datasets matching and fusion. The tool generates an alignment that can be further used to either fuse the two matched datasets or produce a set of links containing the *owl:sameAs* triples. The system takes as input XML files specifying the preprocessing parameters: name reordering, property strings translation, datasets ontological structure, and matching techniques for each kind of resource. The datasets structure and the resources to be matched are described in two files. This descriptions in fact corresponds to a small ontology containing only resources of interest and the properties to be used in the matching process. Another configuration file describes post-processing parameters such as the threshold for generating links, as well the fusion parameters in case of a fusion. The tool works with local copies of the datasets and is implemented in Java.

Zhishi.links. Zhishi.links (Niu et al., 2011) is a general purpose data linking tool using a distributed framework to reduce the time complexity when matching large datasets. Resources are therefore indexed before similarity calculations are performed. Two similarity comparisons are available, based on entities names (RDFS and SKOS labels, aliases) and based on geographical location. Then a semantic matching technique is used to increase the similarity of instances sharing same property-value pairs. Candidates are finally sorted by their similarity value score, using an eventual new computation on their default label to disambiguate candidates having the same score. The system also uses abbreviations list for persons, locations and organizations (i.e., “Jr” for junior).

Serimi. Serimi (Araujo et al., 2011) interlinking tool use a two phases approach. In a first phase, traditional information retrieval strategies are applied to select candidate resource to be linked. More precisely, entity labels of the source dataset are used to search for entities in the target dataset. In a second phase, candidates matches sharing same labels are disambiguated using an algorithm that identifies which of the candidates is more likely to be the correct one. This technique is based on an analysis of resources descriptions, by identifying property sets shared by instances. *Classes of interest* are thus formed. Instances of the same class in the source dataset are finally linked to instances of the same class of interest in the target dataset.

Summary. As we can see in tables 2, 3, and 4, varying data linking techniques are implemented by the studied systems.

Table 2. Systems value matching techniques

	Data Level		Knowledge Level
System	Internal	External	
LN2R	String matching	-	WordNet Synonyms dictionary
ObjectCoref	String matching	-	-
OKKAM ENS	String matching	Translation service	Entity names vocabulary
RKB-CRS	String matching	-	-
LD Mapper	String lookup search	-	-
Silk	String matching Numerical similarity	-	-
LIMES	String matching on metric spaces	-	-
KnoFuss	String matching	-	-
RDF-AI	Fuzzy string matching	Translation service	WordNet Taxonomic distance
Zhishi.links	String matching	-	Abbreviations list
Serimi	String matching	-	-

It seems obvious that internal data level techniques are implemented in every system for value matching as they are the basic operation for comparing instances values. Only two systems use translation services to translate strings before comparing them. This can surely be explained by the fact a large majority of the data available on the Web of data is in English. External knowledge sources are commonly used to compute similarity between instances values.

Table 3. Systems individual matching techniques

	Data Level		Knowledge Level
System	Internal	External	
LN2R	Maximum weighed average	-	sameAs InverseFunctional FunctionalProperty cardinality maxCardinality
ObjectCoref	attribute-based similarity (learned)	Existing links	sameAs InverseFunctional FunctionalProperty cardinality maxCardinality
OKKAM ENS	similarity combination techniques (listed in paper)	-	-
RKB-CRS	domain-specific metrics for citations	Existing links	-
LD Mapper	Sum	-	-
Silk	Many	-	-
LIMES	-	-	-

KnoFuss	Weighted average	sameAs transitive closure	Cardinality Disjointness Functionality
RDF-AI	Weighted average	-	-
Zhishi.links	Geographical distance	-	Functional Inverse functional
Serimi	-	-	-

Most tools use similarity aggregation in order to combine similarity over many attributes of the matched instances, but few tools make use of existing links (data level, external techniques). As the density of links on the Web of data increase, reusing existing links or computing transitivity closures might be useful. It is however crucial to consider links quality in order to prevent the propagation of incorrect links. Three tools only consider the ontology axioms when computing the similarity of two instances. One reason behind this is that few vocabularies on the Web of data actually have these axioms. As more work will be done on increasing vocabularies quality, data linking using ontological axioms will become more and more important.

Table 4. Systems dataset matching techniques

	Data Level	Knowledge Level	
System		Internal	External
LN2R	Iterative similarity	“Schema aware” similarity propagation	-
ObjectCoref	Existing links between web datasets	-	-
OKKAM ENS	-	-	-
RKB-CRS	-	-	-
LD Mapper	-	-	-
Silk	-	-	-
LIMES	-	-	-
KnoFuss	-	Dempster-Shafer similarity propagation	Third-party schema mappings
RDF-AI	Similarity propagation	-	-
Zhishi.links	-	-	-
Serimi	Classes of interest	-	-

Dataset matching techniques are the least used ones. Indeed they rely on the previous set of techniques and thus are the most advanced ones. One tool only makes use of third-party schema mappings. This can be seen as very low given that reconciling schemas greatly facilitates the matching task. Semi automated tools actually solve this problem by implicitly letting the user to specify the schema alignment when configuring the tool for a specific matching task.

Table 5 shows that the tools require different levels of manual input. While most of the tools require manual input, two tools only are fully automated: LN2R and LD Mapper. These two tools can only be used on datasets sharing a same ontology, with LD Mapper being specialized for the MusicOntology. Another tool, ObjectCoref, does not need to be configured but instead takes as input a training set that it will use to find the best appropriated configuration. The other tools require a user specification that varies in syntax, but have similar content: what kind of entities are linked, which properties are compared and which similarity computation methods are used. RKB CRS follows this scheme but includes built-in heuristics making it more efficient for linking university and conferences related datasets.

Table 5. Systems usability and manual input

LN2R	Automated, works only if datasets share the same ontology
ObjectCoref	Needs a training set, then automated
OKKAM ENS	Must be run on one entity type at a time (e.g., Persons, places)
RKB-CRS	A Java class has to be implemented for each pair of datasets
LD Mapper	Automated, works only for datasets described using the MusicOntology
Silk	Link specification language: used techniques, implicit schema alignment, aggregations need to be described
LIMES	Link specification language, Web service and Web GUI
KnoFuss	Dataset description ontology needs to be written; ontology alignments can be reused
RDF-AI	Dataset, techniques, pre- and post- processing configuration files need to be written
Zhishi.links	Information not available
Serimi	Automated, command line parameters

We will next discuss techniques that could be used to make tools more efficient and more automated.

OPEN PROBLEMS AND CONCLUDING REMARKS

In the Semantic Web domain, data linking represents a relatively new direction, which largely follows the research conducted in generic ontology matching and especially in the database research community. Thus, we can say that many open problems currently considered in these areas are also relevant for data linking: for example, such issues as scalability (Rastogi et al., 2011), unreliable data sources (Dong et al., 2009), erroneous data values (Guo et al., 2010), and the need to take links into account when processing queries (Ioannou, Nejdl, et al., 2010). However, the task of semantic data linking has its unique features: the use of ontologies for data structuring and the availability of large volumes of distributed data published using the Linked Data principles. These features lead to several research problems, which are primarily relevant for the data linking task.

As discussed in the previous sections, in the field of data linking a clear and widely accepted definition of the meaning of mappings is still missing. On one side, we have a wide spectrum of different techniques, capable of discovering different kinds of possible links between object descriptions, ranging from weak correspondences to strong relations of identity. On the other side, we still use (or abuse) the *owl:sameAs* relation for representing all these different links, in spite of the original meaning of this specific OWL relation.

Concerning this point, a lot of work is possible towards a finer definition of the relation of identity and a richer vocabulary for representing it. Both UMBEL and SKOS can be seen as useful starting points for the definition of a shared vocabulary concerning identity and mappings. However, covering the range of possible meanings for a mapping is just part of the problem: in fact, we also need to clarify and study the correspondence between the different techniques available for instance matching and the possible meanings of their resulting mappings. To this end, some preliminary work has been proposed by McCusker & McGuinness (2010), where the authors discuss a solution that allows the extraction of isomorphic statements from data without requiring their direct assertion and propose a Identity Ontology for the representation of identity. However, an analytic inquiry about the formal properties of instance matching techniques with respect to identity is still missing.

Currently, published identity links between individuals indicate equivalent instances. However, it is also important to maintain information about pairs of individuals, which represent distinct entities, especially if these individuals have similar descriptions which can be misleading for data linking tools. Providing the means to state and maintain such distinction relations represents another research problem, for which existing solutions can be insufficient: e.g., using explicit *owl:differentFrom* statements between each pair of distinct individuals would lead to very large amount of auxiliary data.

Besides the ones chosen for this work, there are other dimensions of analysis that can taken into account for presenting the data linking problem, such as for example the degree of automation, the human contribution.

Our approach was to concentrate not on the engineering aspects of the tools (e.g., pre-processing, post-processing, programming language, human contribution) but on the underlying techniques. About this point, there are several potentially promising research directions concerning the development of novel matching techniques. We can particularly point out two of them. First, the growth of the Linked Data cloud provides unique possibility to use large volumes of already existing data as information sources. Moreover, the task of data linking itself becomes transformed from the traditional scenario, which focuses on finding sets of mappings between two datasets to the more open task of discovering mappings within a network of many datasets. This makes the dataset matching techniques particularly important. Two examples of tools targeting these issues are idMesh (Cudré-Mauroux et al., 2009) and ObjectCoref (Hu et al., 2011). Second, given the variety of used schema vocabularies, methods able to overcome semantic heterogeneity and deal with information expressed using different ontologies are especially valuable.

With regards to the tools, we see that there is still room for improving their automation. Most tools are semi-automated and require an extensive amount of configuration for each data linking task, while automatic tools work only in predefined domains. We identify three types of input: i) schema alignments, ii) matching techniques, and iii) key properties.

We can envisage three possible improvements to automate existing tools. Schema alignments could be shared using a server, such as the R2R framework (Bizer & Schultz, 2010). By doing so they could be reused for every data linking task involving the aligned schemas. The selection of matching techniques is depending on the kind of data needed to be matched as some techniques are better working than others on specific data types. Information about the techniques to use could be directly attached to the schemas and thus would not need to be specified for each data linking task. Finally, data has to be analyzed in order to specify what properties need to be compared for determining the identity of two instances. This set of discriminative properties ensure that there are no two instances sharing the same combination of values for these properties: in other words, they have the same role as a key in a relational database. Statistical analysis could be used in order to automatically mine such key properties in RDF datasets.

In conclusion, after defining the data linking problem, we have presented a comprehensive survey of techniques helping to solve it. We have classified these techniques according to the criteria of granularity, type of evidence, and source of the evidence. We have presented eleven data linking systems and classified them according to which technique they use. We observed that the studied systems leave room for improvement, particularly at the dataset level of granularity.

REFERENCES

Alexander, K., Cyganiak, R., Hausenblas, M., & Zhao, J. (2009). Describing linked datasets - on the design and usage of void, the “vocabulary of interlinked datasets”. *In Linked data on the web workshop (LDOW ‘09), workshop at 18th international world wide web conference (WWW ‘09)*. Madrid, Spain.

Araujo, S., Vries, A. de, & Schwabe, D. (2011). Serimi results for OAEI 2011. Paper presented at the Ontology matching workshop at ISWC 2011, ontology alignments evaluation initiative (OAEI 2011).

Batini, C., & Scannapieco, M. (2006). *Data quality: Concepts, methodologies and techniques (data-centric systems and applications)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Berners-Lee, T., Hollenbach, J., Lu, K., Presbrey, J., Prud’hommeaux, E., & Schraefel, M. (2008). Tabulator Redux: Browsing and Writing Linked Data. *In Proc. of the WWW Int. workshop on linked data on the web (LDOW 2008)*. Beijing, China.

Bizer, C., & Schultz, A. (2010). The r2r framework: Publishing and discovering mappings on the web. *In 1st international workshop on consuming linked data (COLD 2010)*. Shanghai, China.

Bleiholder, J., & Naumann, F. (2009). Data fusion. *ACM Comput. Surv.*, 41(1), 1-41.

Bouquet, P., Stoermer, H., & Bazzanella, B. (2008). An entity name system (ens) for the semantic web. *In Proceedings of the 5th European semantic web conference on the semantic web: research and applications* (pp. 258–272). Tenerife, Spain.

Bryl, V., Giuliano, C., Serafini, L., & Tymoshenko, K. (2010). Supporting natural language processing with background knowledge: Coreference resolution case. *In 9th International semantic web conference (ISWC 2010)* (pp. 80–95). Shanghai, China.

Castano, S., Ferrara, A., Lorusso, D., Näth, T. H., & Möller, R. (2008). Mapping validation by probabilistic reasoning. *In Proceedings of the 5th European semantic web conference on the semantic web: research and applications* (pp. 170–184). Tenerife, Spain.

Castano, S., Ferrara, A., & Montanelli, S. (2006). Matching Ontologies in Open Networked Systems: Techniques and Applications. *Journal on Data Semantics*, V, 25–63.

Cilibrasi, R., & Vitanyi, P. (2007). The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3), 370–383.

Cohen, W. W., Ravikumar, P., & Fienberg, S. E. (2003). A comparison of string metrics for matching names and records. *In Kdd workshop on data cleaning and object consolidation*.

Cour, T., Srinivasan, P., & Shi, J. (2006). Balanced graph matching. *Advances in Neural Information Processing Systems*, 19, 313– 320.

Cudré-Mauroux, P., Haghani, P., Jost, M., Aberer, K., & Meer, H. de. (2009). idMesh: Graph-based disambiguation of linked data. *In 18th International World Wide Web conference (WWW 2009)* (pp. 591–600). Madrid, Spain.

Dong, X., Berti-Equille, L., & Srivastava, D. (2009). Integrating conflicting data: The role of source dependence. *In 35th International conference on very large databases (VLDB '09)* (pp. 550– 561). Lyon, France.

Dong, X., Halevy, A., & Madhavan, J. (2005). Reference reconciliation in complex information spaces. *In SIGMOD '05: Proceedings of the 2005 ACM SIGMOD International conference on management of data* (pp. 85–96). New York, NY, USA.

Elmagarmid, A., Ipeirotis, P., & Verykios, V. (2007, January). Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1), 1–16.

Euzenat, J., & Shvaiko, P. (2007). *Ontology Matching*. Heidelberg (Germany): Springer-Verlag.

Fellegi, I. P., & Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64(328), 1183– 1210.

Glaser, H., Millard, I., & Jaffri, A. (2008). Rkbexplorer. com: a knowledge driven infrastructure for linked data providers. *In Proceedings of the 5th European semantic web conference on the semantic web: research and applications* (pp. 797–801). Tenerife, Spain.

Gligorov, R., Aleksovski, Z., Kate, W. ten, & Harmelen, F. van. (2007). Using Google distance to weight approximate ontology matches. *In 16th International World Wide Web conference (WWW 2007)* (pp. 767–775). Banff, Canada.

Gracia, J., d'Aquin, M., & Mena, E. (2009). Large scale integration of senses for the Semantic Web. *In 18th International World Wide Web conference (WWW 2009)*. Madrid, Spain.

Gracia, J., & Mena, E. (2008). Matching with CIDER: Evaluation report for the OAEI 2008. *In 3rd ontology matching workshop (OM '08) at the 7th International semantic web conference (ISWC '08)*. Karlsruhe, Germany.

Guarino, N., & Welty, C. (2002). Identity and Subsumption. In *The semantics of relationships: An interdisciplinary perspective* (pp. 111–125). Dordrecht, The Netherlands: Kluver.

Guo, S., Dong, X., Srivastava, D., & Zajac, R. (2010). Record linkage with uniqueness constraints and erroneous values. *In 36th International conference on very large databases (VLDB 2010)* (pp. 417–428). Singapore.

Halpin, H., Hayes, P. J., McCusker, J. P., McGuinness, D. L., & Thompson, H. S. (2010). When owl:sameas isn't the same: An analysis of identity in linked data. In 9th International semantic web conference (ISWC 2010) (pp. 305–320). Shanghai, China.

Hassanzadeh, O., Lim, L., Kementsietsidis, A., & Wang, M. (2009). A declarative framework for semantic link discovery over relational data. In WWW '09: Proceedings of the 18th International conference on World Wide Web (pp. 1101–1102). New York, NY, USA.

Hu, W., Chen, J., & Qu, Y. (2011). A self-training approach for re- solving object coreference on the semantic web. *In Proceedings of the 20th International World Wide Web Conference (WWW 2011)* (pp. 87–96). Hyderabad, India.

Ioannou, E., Nejdl, W., Niederée, C., & Velegrakis, Y. (2010). On-the-fly entity-aware query processing in the presence of linkage. In 36th International conference on very large databases (VLDB 2010) (pp. 429–438). Singapore.

Ioannou, E., Niederée, C., & Nejdl, W. (2008). Probabilistic entity linkage for heterogeneous information spaces. *In Advanced Information Systems Engineering, 20th International Conference (CAiSE 2008)* (pp. 556–570). Montpellier, France.

Ioannou, E., Niederée, C., Velegrakis, Y., Bonvin, N., Mocan, A., Papadakis, G., et al. (2010). *Intelligent entity matching and ranking* (Tech. Rep. D3.1). OKKAM Project.

Jaffri, A., Glaser, H., & Millard, I. (2008). URI disambiguation in the context of Linked Data. *In Proc. of the WWW Int. workshop on linked data on the web (LDOW 2008)*. Beijing, China.

Jean-Mary, Y., Shironoshita, E., & Kabuka, M. (2009). Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3), 235–251.

Köpcke, H., & Rahm, E. (2010). Frameworks for entity matching: A comparison. *Data Knowl. Eng.*, 69, 197–210.

Köpcke, H., Thor, A., & Rahm, E. (2010). Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment*, 3(1-2), 484–493.

Koudas, N., Sarawagi, S., & Srivastava, D. (2006). Record linkage: similarity measures and algorithms. *In Proceedings of the 2006 ACM SIGMOD international conference on Management of data* (pp. 802–803). Chicago, Illinois, USA.

Li, J., Tang, J., Li, Y., & Luo, Q. (2009). RiMOM: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering*, 21(8), 1218–1232.

Lopez, V., Nikolov, A., Fernandez, M., Sabou, M., Uren, V., & Motta, E. (2009). Merging and ranking answers in the Semantic Web: The wisdom of crowds. *In Proceedings of the 4th Asian Se- mantic Web Conference (ASWC 2009)* (pp. 135–152). Shanghai, China.

McCusker, J., & McGuinness, D.(2010). Towards identity in linked data. *In Proceedings of OWL Experiences and Directions Seventh Annual Workshop*. Karlsruhe, Germany.

Meilicke, C., Völker, J., & Stuckenschmidt, H. (2008). Learning disjointness for debugging mappings between lightweight ontologies. *In Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns* (pp. 93–108). Berlin, Heidelberg.

Melnik, S. (2002). Similarity flooding: a versatile graph matching algorithm. *In Proc. 18th International Conference on Data Engineering (ICDE 2002)* (pp. 117–128). San Jose, CA, USA.

Miller, G. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39–41.

Ngomo, A.-C. N., & Auer, S. (2011). LIMES - a time-efficient approach for large-scale link discovery on the web of data. *In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI 2011)*. Barcelona, Spain.

Nikolov, A., d'Aquin, M., & Motta, E. (2011). Unsupervised instance coreference resolution using a genetic algorithm (Tech. Rep. No. kmi-11-02). Knowledge Media Institute, UK.

Nikolov, A., Uren, V., & Motta, E. (2010). Data linking: Capturing and utilising implicit schema-level relations. *In 3rd workshop on linked data on the web (LDOW 2010)*. Raleigh, USA.
Nikolov, A., Uren, V., Motta, E., & Roeck, A. de. (2008a). Integration of semantically annotated data by the KnoFuss architecture. *In 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008)* (pp. 265–274). Acitrezza, Italy.

Nikolov, A., Uren, V., Motta, E., & Roeck, A. de.(2008b). Refining instance coreferencing results using belief propagation. *In 3rd Asian Semantic Web Conference (ASWC 2008)* (pp. 405–419). Bangkok, Thailand.

Niu, X., Rong, S., Zhang, Y., & Wang, H. (2011). Zhishi.links results for oaei 2011. *In Ontology Matching workshop at ISWC 2011, Ontology Alignments Evaluation Initiative (OAEI 2011).* Bonn, Germany.

Noessner, J., Niepert, M., Meilicke, C., & Stuckenschmidt, H. (2010). Leveraging terminological structure for object reconciliation. *In 7th Extended Semantic Web Conference (ESWC 2010)* (pp. 334–348). Heraklion, Crete, Greece.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco: Morgan Kaufmann.

Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4), 334– 350.

Raimond, Y., Sutton, C., & Sandler, M. (2008). Automatic interlinking of music datasets on the semantic web. *In Proceedings of the Linking Data On the Web workshop at WWW 2008 (LDOW 2008)*. Beijing, China.

Rastogi, V., Dalvi, N., & Garofalakis, M. (2011). Large-scale collective entity matching. *Proceedings of the VLDB Endowment*, 4(4), 208–218.

Säis, F., Pernelle, N., & Rousset, M.-C. (2007). L2R: A Logical Method for Reference Reconciliation. *In AAAI-07: Twenty-Second Conference on Artificial Intelligence* (pp. 329–334). Vancouver, BC, Canada.

Säis, F., Pernelle, N., & Rousset, M.-C.(2008). Combining a logical and a numerical method for data reconciliation. *Journal of Data Semantics*, 12, 66–94.

Scharffe, F., Liu, Y., & Zhou, C. (2009). RDF-AI: an architecture for RDF datasets matching, fusion and interlink. *In Workshop on Identity and Reference in Knowledge Representation, IJCAI 2009*. Pasadena, CA, USA.

Schrijver, A. (1998). *Theory of linear and integer programming*. New York, NY, USA: John Wiley & Sons, Inc.

Shafer, G. (1976). *A mathematical theory of evidence*. Princeton, USA: Princeton University Press.

Shenoy, P. P.(1992). Valuation-based systems: a framework for managing uncertainty in expert systems. *In Fuzzy logic for the management of uncertainty* (pp. 83–104). New York, NY, USA: John Wiley & Sons, Inc.

Stoilos, G., Stamou, G., & Kollias, S. (2005). A string metric for ontology alignment. *In 4th International Semantic Web Conference (ISWC 2005)* (pp. 624–637). Galway, Ireland.

Tummarello, G., Delbru, R., & Oren, E. (2007). Sindice.com: Weaving the open linked data. *In 6th International Semantic Web Conference (ISWC/ASWC 2007)* (pp. 552–565). Busan, Korea.

Udrea, O., Getoor, L., & Miller, R. J. (2007). Leveraging data and structure in ontology integration. *In Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (pp. 449–460). Beijing, China.

Volz, J., Bizer, C., & Gaedke, M. (2009). Web of data link maintenance protocol (Technical Report). Frei Universität Berlin.

Volz, J., Bizer, C., Gaedke, M., & Kobilarov, G. (2009). Discovering and maintaining links on the web of data. *In 8th International Semantic Web Conference (ISWC 2009)* (pp. 650–665). Washington, DC, USA.

Winkler, W. (2006). Overview of record linkage and current research directions (Tech. Rep. No. 2006-2). Statistical Research Division. U.S. Census Bureau.

[i] Names of the authors are listed in the alphabetic order

[ii] In this context, we use the term real-world object in order to denote the intended referent of an object description.

[iii] Sometimes, the term *mapping* is used in order to denote the collection of binary relations between object descriptions and the term *mapping rule* is used in order to denote the single correspondence between two object descriptions. This terminology is more common in the field of ontology matching, where mappings often represent more than a simple correspondence between entities, but a transformation rule holding between two entities (Euzenat & Shvaiko, 2007). Another terminological choice is to use the term *alignment* in order to denote the mapping set and the term *correspondence* to denote the mapping.

[iv] See http://umbel.org and http://www.w3.org/2004/02/skos

[v] For an interesting discussion on this point see http://www.mkbergman.com/935/the-nature-of-connectedness-on-the-web (Retrieved October, 2011).

[vi] http://sourceforge.net/projects/simmetrics (Retrieved October 2011)

[vii] http://secondstring.sourceforge.net (Retrieved October 2011)

[viii] http://www.ontologyportal.org (Retrieved October 2011)

[ix] http://www.opencyc.org (Retrieved October 2011)

[x] http://www.w3.org/2004/02/skos (Retrieved October 2011)

[xi] http://www.sameas.org (Retrieved October 2011)

[xii] http://www.linkedmdb.org (Retrieved October 2011)

[xiii] http://dbpedia.org (Retrieved October 2011)

[xiv] http://dbtune.org/musicbrainz (Retrieved October 2011)

[xv] http://watson.kmi.open.ac.uk (Retrieved October 2011)

[xvi] http://www.wikipedia.org (Retrieved October 2011)

[xvii] http://web.expasy.org/docs/userman.html (Retrieved October 2011)

[xviii] http://www.genome.jp/kegg/genes.html (Retrieved October 2011)

[xix] Since the approach is aimed at processing annotations extracted from text, data statements are not considered 100% reliable.

[xx] http://www4.wiwiss.fu-berlin.de/bizer/silk/spec (Retrieved October 2011)

[xxi] http://aksw.org/Projects/limes (Retrieved October 2011)

[xxii] http://limes.aksw.org (Retrieved October 2011)

[xxiii] http://alignapi.gforge.inria.fr/format.html (Retrieved October 2011)