Linking and Building Ontologies of Linked Data

Rahul Parundekar, Craig A. Knoblock, and José Luis Ambite

University of Southern California,
Information Sciences Institute and Department of Computer Science
4676 Admiralty Way, Marina del Rey, CA 90292
{parundek,knoblock,ambite}@isi.edu

Abstract. The Web of Linked Data is characterized by linking structured data from different sources using equivalence statements, such as owl:sameAs, as well as other types of linked properties. The ontologies behind these sources, however, remain unlinked. This paper describes an extensional approach to generate alignments between these ontologies. Specifically our algorithm produces equivalence and subsumption relationships between classes from ontologies of different Linked Data sources by exploring the space of hypotheses supported by the existing equivalence statements. We are also able to generate a complementary hierarchy of derived classes within an existing ontology or generate new classes for a second source where the ontology is not as refined as the first. We demonstrate empirically our approach using Linked Data sources from the geospatial, genetics, and zoology domains. Our algorithm discovered about 800 equivalences and 29,000 subset relationships in the alignment of five source pairs from these domains. Thus, we are able to model one Linked Data source in terms of another by aligning their ontologies and understand the semantic relationships between the two sources.

1 Introduction

The last few years have witnessed a paradigm shift from publishing isolated data from various organizations and companies to publishing data that is *linked* to related data from other sources using the structured model of the Semantic Web. As the data being published on the Web of Linked Data grows, such data can be used to supplement one's own knowledge base. This provides significant benefits in various domains where it is used in the integration of data from different sources. A necessary step to publish data in the Web of Linked Data is to provide links from the instances of a source to other data 'out there' based on background knowledge (e.g. linking DBpedia to Wikipedia), common identifiers (e.g. ISBN numbers), or pattern matching (e.g. names, latitude, longitude and other information used to link Geonames to DBpedia). These links are often expressed by using *owl:sameAs* statements. Often, when such links between instances are asserted, the link between their corresponding concepts is not made. Such conceptual links would ideally help a consumer of the information (agent/human) to model data from other sources in terms of their own knowledge. This problem is widely known as ontology alignment [12], which is a form of schema alignment [16]. The advent of the Web of Linked Data warrants a renewed inspection of these methods.

P.F. Patel-Schneider et al. (Eds.): ISWC 2010, Part I, LNCS 6496, pp. 598–614, 2010.

Our approach provides alignments between classes from ontologies in the Web of Linked Data by examining their linked instances. We believe that providing ontology alignments between sources on the Web of Linked Data provides valuable knowledge in understanding and reusing such sources. Moreover, our approach can provide a more refined ontology for a source described with a simple ontology (like GEONAMES) by aligning it with a more elaborate ontology (like DBPEDIA). Alternatively, by aligning an ontology (like GEOSPECIES) with itself using the same approach, we are able to generate a hierarchy of derived classes, which provide class definitions complimentary to those already existing in the source.

The paper is organized as follows. First, we briefly provide background on Linked Data and describe the domains (geospatial, genetics and zoology) and data sources (LINKEDGEODATA, GEONAMES, DBPEDIA, GEOSPECIES, MGI, and GENEID) that we focus on in this paper. Second, we describe our approach to ontology alignment, which is based on defining *restriction classes* over the ontologies and comparing the extensions of these classes to determine the alignments. Third, we provide an empirical evaluation of the alignment algorithm on five pairs of sources: (LINKEDGEODATA-DBPEDIA, GEONAMES-DBPEDIA, GEOSPECIES-DBPEDIA, MGI-GENEID and GEOSPECIES-GEOSPECIES). Finally, we describe related and future work and discuss the contributions of this paper.

2 Linked Data Background and Sources

In this section, we provide a brief introduction to Linked Data and the three domains and six data sources that we consider.

The Linked Data movement, as proposed by Berners-Lee [6], aims to provide machine-readable connections between data in the Web. Bizer et al. [7] describe several approaches to publishing such Linked Data. Most of the Linked Data is generated automatically by converting existing structured data sources (typically relational databases) into RDF, using an ontology that closely matches the original data source. For example, GEONAMES gathers its data from over 40 different sources and it primarily exposes its data as a flat-file structure[1] that is described with a simple ontology [19]. Such an ontology might have been different if designed at the same time as the collection of the actual data. For example, all instances of GEONAMES have *geonames:Feature* as their only *rdf:type*, however, they could have been more effectively typed based on the *featureClass* and *featureCode* properties (cf. Section 3.1).

The links in the Web of Linked Data make the Semantic Web browsable and, moreover, increase the amount of knowledge by complementing data in a source with existing data from other sources. A popular way of linking data on the Web is the use of *owl:sameAs* links to represent *identity links* [14,8]. Instead of reusing existing URIs, new URIs are automatically generated while publishing linked data and an *owl:sameAs* link is used to state that two URI references refer to the same thing. Halpin et al. [14] distinguish four types of semantics for these links: (1) same thing as but different context, (2) same thing as but referentially opaque, (3) represents, and (4) very similar to.

[1] `http://download.geonames.org/export/dump/readme.txt`

For the purposes of this paper, we refrain from going into the specifics and use the term as asserting equivalence.

In this paper we consider six sources sources from three different domains (geospatial, zoology, and genetics), which are good representatives of the Web of Linked Data:

LINKEDGEODATA is a geospatial source with its data imported from the Open Street Map (OSM) [13] project containing about 2 billion triples. The data extracted from the OSM project was linked to DBPEDIA by expanding on the user created links on OSM to WIKIPEDIA using machine learning based on a heuristic on the combination of type information, spatial distance, and name similarity [4].

GEONAMES is a geographical database that contains over 8 million geographical names. The structure behind the data is the Geonames ontology [19], which closely resembles the flat-file structure. A typical individual in the database is an instance of type *Feature* and has a *Feature Class* (administrative divisions, populated places, etc.), a *Feature Code* (subcategories of *Feature Class*) along with latitude, longitude, etc. associated with it.

DBPEDIA is a source of structured information extracted from WIKIPEDIA containing about 1.5 million objects that are classified with a consistent ontology. Because of the vastness and diversity of the data in DBPEDIA, it presents itself as a hub for links in the Web of Linked Data from other sources [3]. We limit our approach to only the *rdf:type* assertions and info-box triples from DBPEDIA as they provide factual information. LINKEDGEODATA, GEONAMES are both linked to DBPEDIA using the *owl:sameAs* property asserting the equivalence of instances.

GEOSPECIES is an initiative intended to unify biological taxonomies and to overcome the problem of ambiguities in the classification of species.[2] GEOSPECIES is linked to DBPEDIA using the *skos:closeMatch* property.

Bio2RDF's MGI and GENEID. The Bio2RDF project aims at integrating mouse and human genomic knowledge by converting data from bioinformatics sources and publishing this information as Linked Data [5]. The Mouse Genome Informatics (MGI) database contains genetic, genomic, and biological data about mice and rats. This database also contains assertions to a gene in the National Center for Biotechnology Information - Entrez Gene database, which is identified with a unique GeneID.[3] The data from the MGI database and Entrez Gene was triplified and published by Bio2RDF on the Web of Linked Data[4], which we refer to as MGI and GENEID. We align these two sources using the *bio2RDF:xGeneID* link from MGI to GENEID.

3 Ontology Alignment Using Linked Data

An *Ontology Alignment* is "a set of correspondences between two or more ontologies," where a *correspondence* is "the relation holding, or supposed to hold according to a particular matching algorithm or individual, between entities of different ontologies" [12]. *Entities* here, can be classes, individuals, properties, or formulas.

[2] `http://about.geospecies.org/`

[3] `http://www.ncbi.nlm.nih.gov/entrez/query/static/help/genehelp.html`

[4] `http://quebec.bio2rdf.org/download/data/`

Our alignment algorithm uses data analysis and statistical techniques for matching the *classes* of two ontologies using what Euzenat et al. [12] classify as a *common extension comparison* approach for aligning the structure. This approach considers classes from different ontologies that have instances in common, and derives the alignment relationship between the classes based on the set containment relationships between the sets of instances belonging to each of the classes. Our approach is novel in two respects. First, we identify common instances by using the equivalence links in the Web of Linked Data. Specifically, we use the *owl:sameAs* property to link LINKEDGEODATA with DBPEDIA, and GEONAMES with DBPEDIA; the *skos:closeMatch* property to link GEOSPECIES with DBPEDIA,[5] and the *bio2rdf:xGeneID* property to link MGI with GENEID. Second, instead of limiting ourselves to the existing classes in an ontology, we overlay a richer space of class descriptions over the ontology and define alignments over these sets of new classes, as we describe next.

3.1 Restriction Classes

In the alignment process, instead of focusing only on classes defined by *rdf:type*, we also consider classes defined by conjunctions of property value restrictions (i.e, *hasValue* constraints in the Web Ontology Language), which we will call *restriction class*es in the rest of the paper. *Restriction classes* help us identify existing as well as derived set of classes in an ontology. A *restriction class* with only a single constraint on the *rdf:type* property gives us a class already present in the ontology, for example in LINKEDGEODATA the restriction (*rdf:type*=lgd:country) identifies the class Country. Using restrictions also helps us get a refined set of classes when the ontology of the source is rudimentary i.e. when there are little or no specializations of top level classes, as can be seen in the case of GEONAMES. In GEONAMES, the *rdf:type* for all instances is *Feature*. Thus, the ontology contains a single concept. Traditional alignments would then only be between the class *Feature* from GEONAMES and another class from DBPEDIA, for example *Place*. However, instances from GEONAMES have *featureCode* and *featureClass* properties. A restriction on the values of such properties gives us classes that we can effectively align with classes from DBPEDIA. For example, the *restriction class* defined by (*featureCode*=geonames:A.PCLI) (independent political entity) aligns with the class *Country* from DBPEDIA. Our algorithm defines restriction classes from the source ontologies and generates alignments between such restrictions classes using subset or equivalence relationships.

The space of *restriction classes* is simply the powerset of distinct property-value pairs occurring in the ontology. For example assume that the GEONAMES source had only three properties: *rdf:type*, *featureCode* and *featureClass*; and the instance *Saudi Arabia* had as corresponding values: geonames:Feature, geonames:A.PCLI, and geonames:A. Then this instance belongs to the *restriction class* defined by (*rdf:type*=geonames:Feature & *featureClass*=geonames:A). The other elements of the powerset also form such restriction classes as shown in Figure 1. It is evident that in order to consider all *restriction classes*, the algorithm would be exponential. We thus need some preprocessing that eliminates those properties that are not useful.

[5] Based on the 'Linked Open Data Cloud Connections' section in `http://about.geospecies.org/`

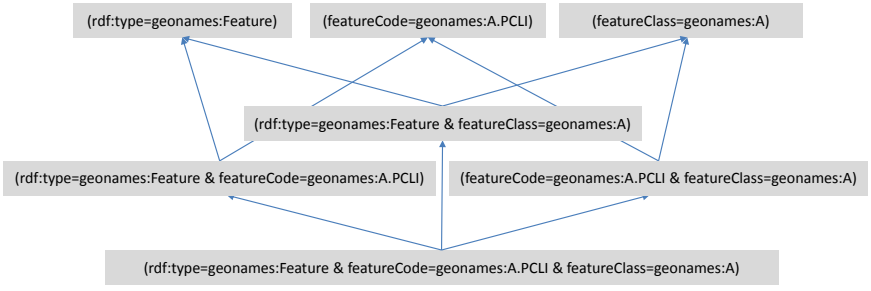

Fig. 1. Hierarchy showing how restriction classes are built

3.2 Pre-processing of the Data

Before we begin exploring alignments, we perform a simple pre-processing on the input sources in order to reduce the search space and optimize the representation. First, for each pair of sources that we intend to align, we only consider instances that are actually linked. For example, instances from DBPEDIA not relevant to alignments in the geospatial domain (like People, Music Albums, etc.) are removed. This has the effect of removing some properties from consideration. For example, when considering the alignment of DBPEDIA to GEONAMES, the *dbpedia:releaseDate* property is eliminated since the instances of type album are eliminated.

Second, in order to reduce the space of alignment hypotheses, we remove properties that cannot contribute to the alignment. Inverse functional properties resemble foreign keys in databases and identify an instance uniquely. Thus, if a *restriction class* is constrained on the value of an inverse functional property, it would only have a single element in it and not be useful. For example, consider the *wikipediaArticle* property in GEONAMES, which links to versions of the same article in WIKIPEDIA in different languages. The GEONAMES instance for the country Saudi Arabia[6] has links to 237 different articles. Each of these, in turn, however could be used to identify only *Saudi Arabia*. Similarly, in LINKEDGEODATA the 'georss:point' property from the 'http://www.georss.org/georss/' namespace contains a String representation of the latitude and longitude and would tend to be an inverse functional property. On the other hand, the *a*ddr:country property in LINKEDGEODATA has a *range* of 2-letter country codes that can be used to group instances into useful restriction classes.

Third, we transform the instance data of a source into a tabular form, which allows us to load the data in a relational database and process it more efficiently. Specifically, each instance is represented as a row in a table, each property occurring in the ontology is a column, and the instance URI is the key. For example, the table for GEONAMES contains 11 columns not including the identifier. We call this tuple representation of an instance a *vector*. In cases of multivalued properties, the row is replicated in such a way that each cell contains a single value but the number of rows equals the number of multiple values. Each new row however, is still identified with the same URI, thus

[6] `http://sws.geonames.org/102358/about.rdf`

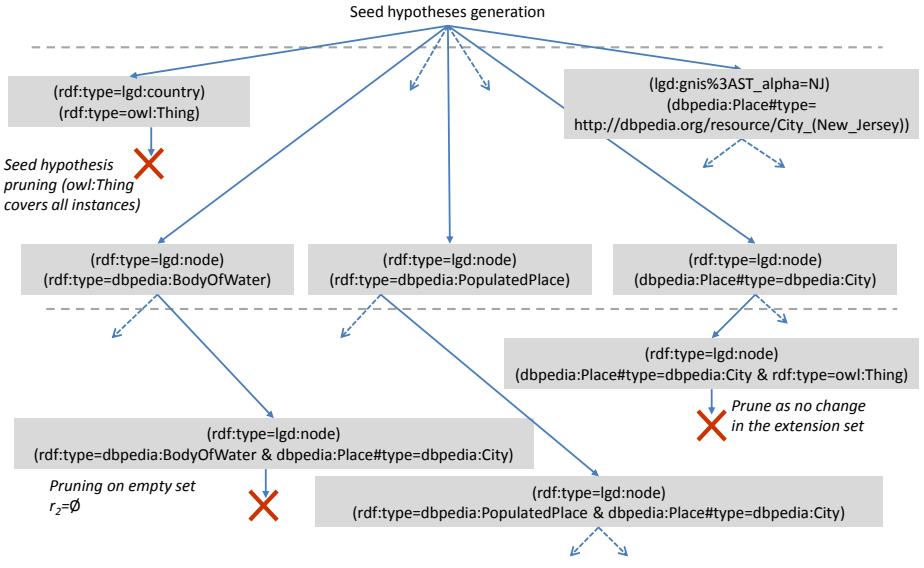

Fig. 2. Exploring and pruning the space of alignments

retaining the number of distinct individuals. In general, the total number of rows for each individual is the product of cardinalities of the value sets for each of its properties.

From these individual vectors, we then perform a join on the equivalence property (e.g. *owl:sameAs* property from LINKEDGEODATA to DBPEDIA) such that we get a combination of vectors from both ontologies. We call this concatenation of two vectors an *instance pair*.

3.3 Searching the Space of Ontology Alignments

An *alignment hypothesis* is a pair of restriction classes, one from each of the ontologies under consideration. The space of alignment hypotheses is combinatorial, thus our algorithm exploits the set containment property of the hypotheses in a top-down fashion along with several pruning features to manage the search space.

We describe the search algorithm that we use to build the alignments by example. Figure 2 shows a small subset of the search space, as explored by this algorithm while aligning LINKEDGEODATA with DBPEDIA. Each gray box represents a candidate hypothesis where the first line within it is the *restriction class* from the first source(O_1) and the second line is the *restriction class* from the second source (O_2). The levels in the exploration space, denoted by dashed horizontal lines, separate alignments where the one from a lower level contains a *restriction class* with one extra property-value constraint than its parent alignment (that is, it is a subclass by construction).

We first seed the space by computing all alignment hypotheses with a single property-value pair from each ontology, that is $[(p_i^1 = v_j^1)(p_k^2 = v_l^2)]$, as shown at the top of Figure 2. There are $O(n^2m^2)$ seed hypotheses, where n is the larger of the number of

properties in each source, and m is the maximum number of distinct values for any property. Then, we explore the hypotheses space by using a depth-first search. At each level we choose a property and add a property-value constraint to one of the *restriction classes* and thus explore all specializations. The *instance pairs* that support the new hypothesis are obtained by restricting the set of *instance pairs* of the current hypothesis with the additional constraint. In Figure 2, while adding a new constraint '*dbpedia:Place#type*=dbpedia:City' to the restriction (*rdf:type*=dbpedia:PopulatedPlace) while aligning it with (*rdf:type*=lgd:node), we take the intersection of the set of identifiers covered by [(*rdf:type*=dbpedia:PopulatedPlace) (*rdf:type*=lgd:node)] with the set of instances in DBPEDIA that have a value of 'dbpedia:City' for the property 'dbpedia:Place#type'.

Our algorithm prunes the search space in several ways. First, we prune those hypotheses with a number of supporting instance pairs less than a given threshold. For example, the hypothesis [(*rdf:type*=lgd:node) (*rdf:type*=dbpedia:BodyOfWater & *dbpedia:Place#type*=dbpedia:City)] is pruned since it has no support.

Second, we prune a seed hypothesis if either of its constituent *restriction classes* covers the entire set of instances from one of the sources, then the algorithm does not search children of this node, because the useful alignments will appear in another branch of the search space. For example, in the alignment between (*rdf:type*=lgd:country) from LINKEDGEODATA and (*rdf:type*=owl:Thing) from DBPEDIA in Figure 2, the *restriction class* (*rdf:type*=owl:Thing) covers all instances from DBPEDIA. The alignment of such a seed hypothesis will always be a subset relation. Moreover, each of its child hypotheses can also be explored through some other hypotheses that does not contain the non-specializing property-value constraint. For example, our algorithm will explore a branch with [(*rdf:type*=lgd:country) (*dbpedia:Place#type*=dbpedia:City)], where the restriction class from the second ontology actually specializes in the extensional sense (*rdf:type*=owl:Thing).

Third, if the algorithm constrains one of the restriction classes of an hypothesis, but the resulting set of *instance pairs* equals the set from the current hypothesis, then it means that adding the constraint did not really specialize the current hypothesis. Thus, the new hypothesis is not explored. Figure 2 shows such pruning when the constraint (*rdf:type*=owl:Thing) is added to the alignment [(*rdf:type*=lgd:node) (*dbpedia:Place#type* =dbpedia:City)].

Fourth, we prune hypotheses $[r_1 r_2]$ where $r_1 \cap r_2 = r_1$ as shown in Figure 3(a). Imposing an additional restriction on r_1 to form r_1' would not provide any immediate specialization as the resultant hypothesis could be inferred from $r_1' \subset r_1$ and the current hypothesis. The same holds for the symmetrical case $r_1 \cap r_2 = r_2$.

Finally, to explore the space systematically the algorithm specializes the restriction classes in a lexicographic order. By doing this, we prune symmetric cases as shown by Figure 3(b). The effect of lexicographic selection of the property can also be seen in Figure 2 when the hypotheses [(*rdf:type*=lgd:node) (*rdf:type*=dbpedia:PopulatedPlace & *dbpedia:Place#type*=dbpedia:City)] is not explored through the hypothesis [(*rdf:type*= lgd:node) (*dbpedia:Place#type*=dbpedia:City)].[7]

[7] Note that the pruning from Figure 3(a) & (b) is not explicitly depicted in Figure 2.

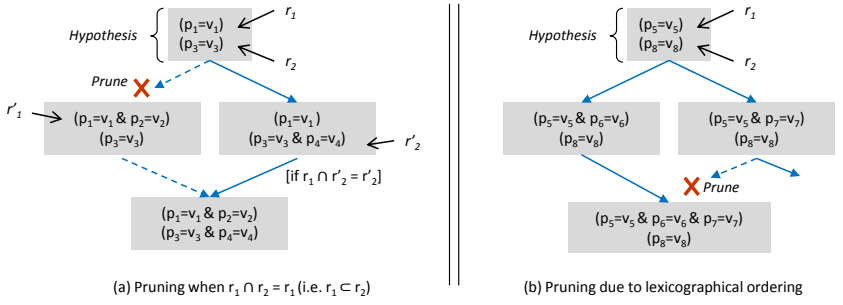

Fig. 3. Pruning the hypotheses search space

3.4 Scoring Alignment Hypotheses

After building the hypotheses, we score each hypothesis to assign a degree of confidence for each alignment. Figure 4 illustrates the instance sets considered to score an alignment. For each hypothesis, we find the instances belonging to the *restriction class* r_1 from the first source and r_2 from the second source. We then compute the *image* of r_1 (denoted by $I(r_1)$), which is the set of instances from the second source that form *instance pairs* with instances in r_1, by following the *owl:sameAs* links. The dashed lines in the figure represent these *instance pairs*. All the pairs that match both restrictions r_1 and r_2 also support our hypothesis and thus are equivalent to the *instance pairs* corresponding to instances belonging to the intersection of the sets r_2 and $I(r_1)$. This set of *instance pairs* that support our hypothesis is depicted as the shaded region. We can now capture subset and equivalence relations between the *restriction classes* by set-containment relations from the figure. For example, if the set of *instance pairs* identified by r_2 are a subset of $I(r_1)$, then the set r_2 and the shaded region would be entirely contained in the $I(r_1)$.

We use two metrics *P* and *R* to quantify these set-containment relations. Figure 5 summarizes these metrics and also the different cases of intersection. In order to allow a certain margin of error induced by the dataset, we are lenient on the constraints and use the relaxed versions *P'* and *R'* as part of our scoring mechanism. For example, consider the alignment between the *restriction class* (*lgd:gnis%3AST_alpha*=NJ) from LINKEDGEODATA to the restriction (*dbpedia:Place#type*=http://dbpedia.org/resource/City_(New_Jersey)) from DBPEDIA shown in Figure 2. Based on the extension sets, our algorithm finds $|I(r_1)| = 39$, $|r_2| = 40$ and $|I(r_1) \cap r_2| = 39$. The value of R' therefore is 0.97 and that of P' is 1.0. Based on our margins, we hence assert the relation of the alignment as equivalent in an extensional sense.

3.5 Eliminating Implied Alignments

From the result set of alignments that pass our scoring thresholds, we need to only keep those that are not implied by other alignments. We hence perform a transitive reduction based on containment relationships to remove the implied alignments. Figure 6 explains

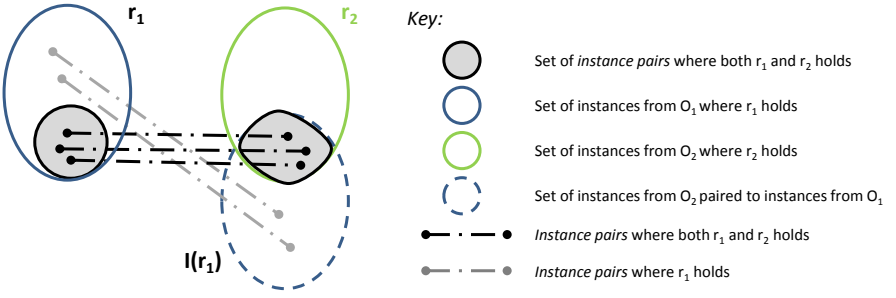

Fig. 4. Scoring of a Hypothesis

Set Representation	Relation	$P = \frac{\lvert I(r_1) \cap r_2 \rvert}{\lvert r_2 \rvert}$	$R = \frac{\lvert I(r_1) \cap r_2 \rvert}{\lvert r_1 \rvert}$	P'	R'
	Disjoint	= 0	= 0	≤ 0.01	≤ 0.01
	$r_1 \subset r_2$	< 1	= 1	> 0.01	≥ 0.90
	$r_2 \subset r_1$	= 1	< 1	≥ 0.90	> 0.01
	$r_1 = r_2$	= 1	= 1	≥ 0.90	≥ 0.90
	Not enough support	0 < P < 1	0 < R < 1	0.01 < P' < 0.90	0.01 < R' < 0.90

Fig. 5. Metrics

these reductions, where alignments between r_1 and r_2 and between r_1' and r_2 are at different levels in the hierarchy such that r_1' is a subclass of r_1 by construction (i.e., by conjoining with an additional property-value pair). Figure 6(a) through (i) depict the combinations of the equivalence and containment relations that might occur in the alignment result set. Solid arrows depict these containment relations. Arrows in both directions denote an equivalence of the two classes.

A typical example of the reduction is Figure 6(e) where the result set contains a relation such that $r_1 \subset r_2$ and $r_1' \subset r_2$. Based on the implicit relation $r_1' \subset r_1$, the relation $r_1' \subset r_2$ can be eliminated (denoted with a cross). Thus, we only keep the relation $r_1 \subset r_2$ (denoted with a check). The relation $r_1 \subset r_2$ could alternatively be eliminated but instead we choose to keep the simplest alignment and hence remove $r_1' \subset r_2$. Other such transitive relations and their reductions are depicted with a 'T' in box on the bottom-right corner.

Another case can be seen in Figure 6(d) where the subsumption relationships found in the alignment results can only hold if all the three classes r_1, r_1' and r_2 are equivalent. These relations have a characteristic cycle of subsumption relationships. We hence need to correct our existing results by converting the subset relations into equivalences. This is depicted by an arrow with a dotted line in the figure. Other similar cases can be seen in Figure 6(a), (c) and (f) where the box on the bottom-right is has a 'C' (cycle).

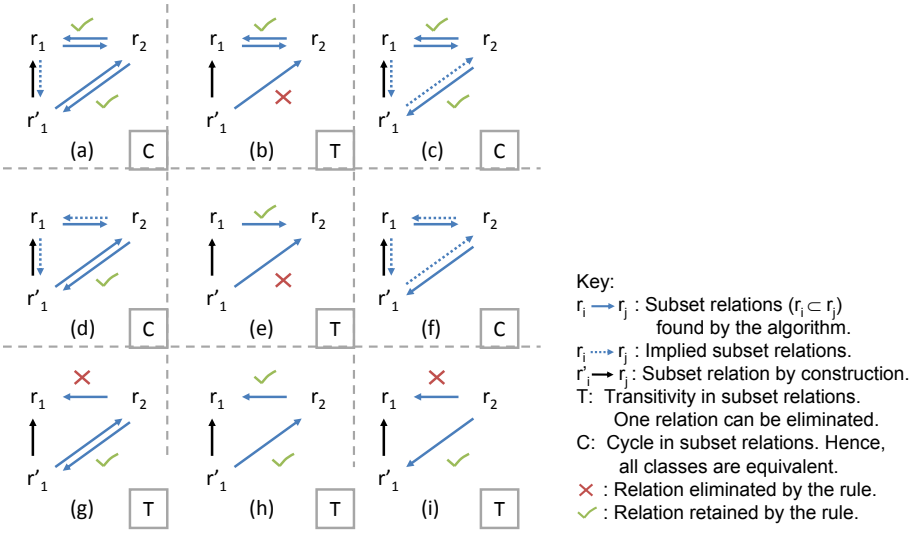

Fig. 6. Eliminating Implied Alignments

In such cases, we order the two equivalences such that the one with more support is said to be a 'better' match than the other (i.e. if $|I(r_1) \cap (r_2)| > |I(r'_1) \cap (r_2)|$, then $r_1 = r_2$ is a better match than $r'_1 = r_2$). The corrections in the result alignments based on transitive reductions may induce a cascading effect. Hence our algorithm applies the 'C' rules shown in Figure 6(a), (c), (d), (f) to identify equivalences until quiescence. Then it applies the 'T' rules to eliminate hypotheses that are not needed.

In sources like DBPEDIA an instance may be assigned multiple *rdf:type*s with values belonging to a single hierarchy of classes in the source ontology. This results in multiple alignments where relations were found to be implied based on the *rdf:type* hierarchy. Such alignments were also considered as candidates for cycle correction, equivalence ordering and elimination of implied subsumptions. We used the ontology files (RDFS/OWL) provided by GEONAMES, LINKEDGEODATA, DBPEDIA AND GEOSPECIES as the source for the ontologies.

4 Empirical Evaluation

We evaluate our algorithm on the domain and sources described in Section 2. Table 1 shows the number of properties and instances in the original sources. For example, LINKEDGEODATA has 5,087 distinct properties and 11,236,351 instances.[8]

As described in Section 3.2, we consider only linked instances and remove properties that cannot generate useful restriction classes. This reduced dataset contains instances that reflect the practical usage of the equivalence links and properties relevant to the domain. In LINKEDGEODATA, most of the instances coming from OSM have a rudimentary type information (classified as 'lgd:node' or 'lgd:way') and are not linked to any

[8] Data and results available at:
`http://www.isi.edu/integration/data/LinkedData`

Table 1. Properties and instances in the original sources

Source	# properties	# instances
LinkedGeoData	5087	11236351
DBpedia	1043	1481003
Geonames	17	6903322
Geospecies	173	98330
MGI	24	153646
GeneID	32	4153014

instance from DBPEDIA. DBPEDIA similarly has instances not linked to LINKEDGEODATA and they were removed as well.

Table 2 shows the results of pre-processing on the source pairs. The table lists the number of properties and instances retained in either sources, the count of the number of combinations of the *vectors* as a result of the join, and the count of the distinct *instance pairs* as identified by the concatenation of their respective URIs. Our algorithm processed this set of *instance pairs* for each source pair and generated alignments that have a minimum support level of 10 *instance pairs*.

Table 2. Generation of instance pairs in pre-processing

Source 1	# properties after elimination	# instances after reduction	Source 2	# properties after elimination	# instances after reduction	# *vector* combinations	# distinct *instance pairs*
LinkedGeoData	63	23594	DBpedia	16	23632	329641	23632
Geonames	5	71114	DBpedia	26	71317	459716	71317
Geospecies	31	4997	Dbpedia	13	4982	289967	4998
MGI	7	31451	GeneID	4	47491	829454	47557
Geospecies	22	48231	Geospecies	22	48231	771690	48231

The alignment results after eliminating implied alignments, as described in Section 3.5, are shown in Table 3. The table shows the two sources chosen for the alignment and the count of the hypotheses classified as equivalent, $r_1 \subset r_2$ and $r_2 \subset r_1$ both before and after elimination.[9] Even though our algorithm provides for the correction and cascading of mislabeled equivalence relations, for all the source pairs that we considered for alignment, such corrections did not arise. The number of equivalences that our algorithm finds can be seen in Table 3 along with the count of equivalences that were labeled as the best match in a hierarchy of equivalence relations. The procedure for elimination of implied relations further prunes the results and helps the system focus on the most interesting alignments. For example, in linking LINKEDGEODATA to DBPEDIA, the 2528 ($r_1 \subset r_2$) relations were reduced to 1837 by removing implied subsumptions. Similarly, in aligning GEOSPECIES with itself, we found 188 equivalence relations, 94 of which were unique due to the symmetrical nature of the hypotheses.

Since the subset and equivalence relationship our algorithm finds are based on extensional reasoning, they hold by definition. However, in the remainder of this section we

[9] The counts of any of the containment relations in the table do not include the logically implied relations within the same source, that is, when r_1' is a subset of r_1 by construction.

Table 3. Alignment results

Source 1 (O_1)	Source 2 (O_2)	#($r_1 = r_2$) total	#($r_1 = r_2$) best matches	#($r_1 \subset r_2$) before	#($r_1 \subset r_2$) after	#($r_2 \subset r_1$) before	#($r_2 \subset r_1$) after
LinkedGeoData	DBpedia	158	152	2528	1837	1804	1627
Geonames	DBpedia	31	19	809	400	1384	1247
Geospecies	DBpedia	509	420	9112	2294	6098	4455
MGI	GeneID	10	9	2031	1869	3594	2070
Geospecies	Geospecies	94	88	1550	1201	-	-

show some examples of the alignments discovered and discuss whether the extensional subset relationships correspond to the intuitive intensional interpretation. As we use an extensional approach as opposed to an intensional one, the results reflect the practical nature of the links between the datasets and the instances in these sources.

Table 4 provides an assessment of the experimental results by selecting some interesting alignment examples from the five source pairs. For each alignment, the table depicts the restrictions from the two sources, the values of the metrics used for hypotheses evaluation (P' and R'), the relation, and the support for that relation.

We refer to the row numbers from Table 4 as a shorthand for the alignments. For example alignment 1 refers to the alignment between the *restriction class* (*rdf:type*=lgd:node) from LINKEDGEODATA and the class (*rdf:type*=owl:Thing) from DBpedia classified as an equivalent relation. Alignments 1, 2, 3 and 5 are the simplest alignments found by our algorithm as they are constrained on values of only the *rdf:type* property. However, we are also able to generate alignments like 4, as shown in Figure 2. GEONAMES has a rudimentary ontology comprised of only a single *Feature* concept. Hence alignments between the restriction classes prove to be more useful. Alignments 6 and 7 suggest that such restrictions from GEONAMES are equivalent to existing concepts in DBPEDIA. Our algorithm is thus able to build a richer set of classes for GEONAMES. This ontology building can also be observed in GEOSPECIES in alignment 12. A more complicated and interesting set of relations is also found in alignments 8, 15, 17, 18, 20 and 22. For example, in alignment 8, pointing a web browser to 'http://sws.geonames.org/3174618/' confirms that for any instance in GEONAMES that has this URI as a parent feature, would also belong to the region of 'Lombardy' in DBPEDIA. In a similar way, 20 provides an alternate definition for a *restriction class* with another class in the same ontology and thus build complimentary descriptions to existing classes and thus reinforce it.

The alignments closely follow the ontological choices of the sources. For example, we could assume that alignment 11, mapping 'geonames:featureCode=T.MT' (Mountain) to 'rdf:type=dbpedia:Mountain', should be equivalent. Closer inspection of the GEONAMES dataset shows, however, that there are some places with *Feature Codes* like 'T.PK' (Peak), 'T.HLL' (Hill), etc. from GEONAMES whose corresponding instances in DBPEDIA are all typed 'dbpedia:Mountain'. This implies that the interpretation of the concept 'Mountain' is different in both the sources and only a subset relation holds. Alignments 16, 19 and 21 also express a similar nature of the classes. As our results follow the data in the sources, incompleteness in the data reflects closely on the alignments generated. Alignment 9 suggests *Schools* from GEONAMES is extensionally equivalent *Educational Institutions*. It should naturally follow that *Schools in the US* be a subset

Table 4. Example alignments from the LINKEDGEODATA-DBPEDIA, GEONAMES-DBPEDIA, GEOSPECIES-DBPEDIA, MGI-GENEID & GEOSPECIES-GEOSPECIES datasets

#	LINKEDGEODATA *restriction*	DBPEDIA *restriction*	*P'*	*R'*	Relation	$\lvert I(r_1) \cap r_2 \rvert$
1	rdf:type=lgd:node	rdf:type=owl:Thing	97.27	99.99	$r_1 = r_2$	22987
2	rdf:type=lgd:aerodrome	rdf:type=dbpedia:Airport	90.94	100	$r_1 = r_2$	251
3	rdf:type=lgd:island	rdf:type=dbpedia:Island	90.81	99.44	$r_1 = r_2$	178
4	lgd:gnis_%3AST_alpha=NJ	dbpedia:Place#type= http://dbpedia.org/resource/City_(New_Jersey)	100	97.5	$r_1 = r_2$	39
5	rdf:type=lgd:village	rdf:type=dbpedia:PopulatedPlace	67.3	98.71	$r_1 \subset r_2$	14391
#	GEONAMES *restriction*	DBPEDIA *restriction*	*P'*	*R'*	Relation	$\lvert I(r_1) \cap r_2 \rvert$
6	geonames:featureClass=geonames:P	rdf:type=dbpedia:PopulatedPlace	91.07	96.7	$r_1 = r_2$	54927
7	geonames:featureClass=geonames:H	rdf:type=dbpedia:BodyOfWater	98.49	91.88	$r_1 = r_2$	1959
8	geonames:parentFeature=http://sws.geonames.org/3174618/	dbpedia:City_region=http://dbpedia.org/resource/Lombardy	99.91	91.2	$r_1 = r_2$	1245
9	geonames:featureCode=geonames:S.SCH	rdf:type=dbpedia:EducationalInstitution	92.45	94.52	$r_1 = r_2$	380
10	geonames:featureCode=geonames:S.SCH & geonames:inCountry=geonames:US	rdf:type=dbpedia:EducationalInstitution	91.72	94.72	$r_1 = r_2$	377
11	geonames:featureCode=geonames:T.MT	rdf:type=dbpedia:Mountain	78.4	96.8	$r_1 \subset r_2$	1728
#	GEOSPECIES *restriction*	DBPEDIA *restriction*	*P'*	*R'*	Relation	$\lvert I(r_1) \cap r_2 \rvert$
12	geospecies:inKingdom=http://lod.geospecies.org/kingdoms/Aa	rdf:type=dbpedia:Animal	99.96	99.96	$r_1 = r_2$	3029
13	geospecies:hasOrderName=Lepidoptera	dbpedia:order=http://dbpedia.org/resource/Lepidoptera	100	99.42	$r_1 = r_2$	344
14	geospecies:hasOrderName=Lepidoptera	dbpedia:kingdom=http://dbpedia.org/resource/Animal & dbpedia:order=http://dbpedia.org/resource/Lepidoptera	100	97.68	$r_1 = r_2$	338
15	geospecies:hasGenusName=Falco	dbpedia:genus=http://dbpedia.org/resource/Falcon	100	90.9	$r_1 = r_2$	10
16	geospecies:hasOrderName=Primates	dbpedia:order=http://dbpedia.org/resource/Primates	100	40.22	$r_2 \subset r_1$	35
#	MGI *restriction*	GENEID *restriction*	*P'*	*R'*	Relation	$\lvert I(r_1) \cap r_2 \rvert$
17	bio2rdf:subType=Pseudogene	bio2rdf:subType=pseudo	93.76	93.56	$r_1 = r_2$	5971
18	bio2rdf:subType=Pseudogene & mgi:genomeStart=17	geneid:chromosome=17 & bio2rdf:subType=pseudo	91.49	94.38	$r_1 = r_2$	269
19	bio2rdf:chromosomePosition=-1.00 & mgi:genomeStart=4	geneid:chromosome=4 & bio2rdf:subType=pseudo	97.07	14.79	$r_2 \subset r_1$	332
#	GEOSPECIES *restriction*	GEOSPECIES *restriction*	*P'*	*R'*	Relation	$\lvert I(r_1) \cap r_2 \rvert$
20	geospecies:hasKingdomName=Animalia	geospecies:inKingdom=http://lod.geospecies.org/kingdoms/Aa	91.99	100	$r_1 = r_2$	563
21	geospecies:hasClassName=Insecta	geospecies:inClass= http://lod.geospecies.org/bioclasses/aQado	87.83	100	$r_1 \subset r_2$	195
22	geospecies:inFamily= http://lod.geospecies.org/families/amTJ9	geospecies:hasSubfamilyName=Sigmodontinae	100	37.03	$r_2 \subset r_1$	10

of *Educational Institutions*. However, as there are only 3 other *Schools* (outside the US), extensionally these classes are very close, as shown by alignment 10. This example illustrates that reasoning extensionally actually provides additional insight on the relationship between the sources. Alignments 13 and 14 show two equivalent alignments that have different support due to missing assertions in one of the ontologies (the property *dbpedia:kingdom* for all moths and butterflies).

Our approach makes an implicit 'closed-world' assumption in using the instances of a class to determine the relationships between the classes in different sources. We believe that this is an important feature of our approach in that it allows one to understand the relationships in the actual linked data and their corresponding ontologies. The alignments generated can be readily used for modeling and understanding the sources since we are modeling what the sources actually contain as opposed as to what an ontology disassociated from the data appears to contain based on the class name or description. Moreover, even if we delve into the open-world assumption of data, it would be very difficult to categorize the missing instances as either: (1) yet unexplored, (2) explored but purposefully classified as not belonging to the dataset, or (3) explored but not included in the dataset by mistake. Hence, our method provides a practical approach to understanding the relationships between sources.

In summary, our algorithm is able to find a significant number of interesting alignments, both equivalent and subset relationships, as well as build and refine the ontologies of real sources in the Web of Linked Data.

5 Related Work

There is a large body of literature on ontology matching [12]. Ontology matching has been performed based on *terminological* (e.g. linguistic and information retrieval techniques [11]), *structural* (e.g. graph matching [15]), and *semantic* (e.g. model-based) approaches or their combination. The FCA-merge algorithm [18] uses extensional techniques over common instances between two ontologies to generate a concept lattice in order to merge them and, thus, align them indirectly. This algorithm, however, relies on a domain expert (a user) to generate the merged ontology and is based on a single corpus of documents instead of two different sources, unlike our approach. A strong parallel to our work is found in Duckham et al. [10], which also uses an extensional approach for fusion and alignment of ontologies in the geospatial domain. The difference in our approach in comparison to their work (apart from the fact that it predates Linked Data) is that while their method fuses ontologies and aligns only existing classes, our approach is able to generate alignments between classes that are derived from the existing ontology by imposing restrictions on values of any or all of the properties not limited to the class *type*. The GLUE system [9] also uses an instance-based similarity approach to find alignments between two ontologies. It uses the labels of the classes that a concept belongs to along with the textual content of the attribute values of instances belonging to that concept to train a classifier and then uses it to classify instances of a concept from the other ontology as either belonging to the first concept or not. Similarly, it also tries to classify the concepts in the other direction. GLUE then hypothesizes alignments based on the probability distributions obtained from the classifications. Our approach,

instead, relies on the links already present in the Web of Linked Data, which in some cases uses a much more sophisticated approach for finding instance equivalences.

Most of the work in information integration within the Web of Linked Data is in instance matching as explained in Bizer et al. [7]. Raimond et al. [17] use string and graph matching techniques to interlink artists, records, and tracks in two online music datasets (Jamendo and MusicBrainz) and also between personal music collections and the MusicBrainz dataset. Our approach solves a complimentary piece of the information integration problem on the Web of Linked Data by aligning ontologies of linked data sources. Schema matching in the Web of Linked Data has also been explored by Nikolov et al. [2], who use existing instance and schema-level evidence of Linked Data to augment instance mappings in those sources. First, instances from different sources are clustered together by performing a transitive closure on *owl:sameAs* links such that all instances in a cluster are equivalent. Class re-assignment is then performed by labeling each instance with all the other classes in the same cluster. Second, a similarity score is computed based on the size of the intersection sets and classes are labeled as equivalent. Finally, more equivalence links are generated based on the new class assignments. Our approach differs from this in the sense that, first, the class re-assignment step increases the coverage of a class. Such an assumption in aligning schemas would bias the extensional approach as it modifies the original extension of a class. Second, only existing classes are explored for similarity in that work and thus faces severe limitations with rudimentary ontologies like GEONAMES, where our approach performs well as it considers restriction classes.

6 Conclusion

The Web of Linked Data contains linked instances from multiple sources without the ontologies of the sources being themselves linked. It is useful to the consumers of the data to define the alignments between such ontologies. Our algorithm generates alignments, consisting of conjunctions of *restriction classes*, that define subsumption and equivalence relations between the ontologies. This paper focused on automatically finding alignments between the ontologies of geospatial, zoology and genetics data sources and building such ontologies using an extensional technique. However, the technique is general and can be applied to other Web of Linked Data data sources.

In our future work, we plan to improve the scalability of our approach, specifically, improve the performance of the algorithm that generates alignment hypotheses by using a more heuristic exploration of the space of alignments. The sizes of the sources in this paper were quite large (on the order of thousands of instances after preprocessing). Although we have fixed a minimum support size of ten instance pairs for a hypothesis, the effectiveness of the extensional approach needs to be verified when the sources are small (number of instances in the order of hundreds or less). We also plan to explore the integration of this work with our previous work on automatically building models of sources [1]. Linking the data from a newly discovered source with a known source already linked to an ontology will allow us to more accurately determine the classes of the discovered data. Finally, we plan to apply our alignment techniques across additional domains and to pursue in depth alignments in biomedical Linked Data.

Acknowledgements

This work was supported in part by the NIH through the following NCRR grant: the Biomedical Informatics Research Network (1 U24 RR025736-01), and in part by the Los Angeles Basin Clinical and Translational Science Institute (1 UL1 RR031986-01).

References

1. Ambite, J.L., Darbha, S., Goel, A., Knoblock, C.A., Lerman, K., Parundekar, R., Russ, T.: Automatically constructing semantic web services from online sources. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 17–32. Springer, Heidelberg (2009)
2. Andriy Nikolov, V.U., Motta, E.: Data Linking: Capturing and Utilising Implicit Schema Level Relations. In: International Workshop on Linked Data on the Web, Raleigh, North Carolina (2010)
3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
4. Auer, S., Lehmann, J., Hellmann, S.: LinkedGeoData: Adding a Spatial Dimension to the Web of Data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 731–746. Springer, Heidelberg (2009)
5. Belleau, F., Tourigny, N., Good, B., Morissette, J.: Bio2RDF: A Semantic Web atlas of post genomic knowledge about human and mouse, pp. 153–160. Springer, Heidelberg (2008)
6. Berners-Lee, T.: Design Issues: Linked Data (2009), `http://www.w3.org/DesignIssues/LinkedData.html`
7. Bizer, C., Cyganiak, R., Heath, T.: How to publish linked data on the web (2007), `http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/`
8. Ding, L., Shinavier, J., Finin, T., McGuinness, D.L.: owl: sameAs and Linked Data: An Empirical Study. In: Second Web Science Conference, Raleigh, North Carolina (2010)
9. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Ontology matching: A machine learning approach. In: Handbook on Ontologies, pp. 385–516 (2004)
10. Duckham, M., Worboys, M.: An algebraic approach to automated geospatial information fusion. International Journal of Geographical Information Science 19(5), 537–557 (2005)
11. Euzenat, J.: An API for Ontology Alignment. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 698–712. Springer, Heidelberg (2004)
12. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg (2007)
13. Haklay, M.M., Weber, P.: OpenStreetMap: user-generated street maps
14. Halpin, H., Hayes, P.J.: When owl: sameAs isn't the same: An analysis of identity links on the semantic web. In: International Workshop on Linked Data on the Web, Raleigh, North Carolina (2010)
15. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: International Conference on Data Engineering, San Jose, California, pp. 117–128 (2002)

16. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal 10(4), 334–350 (2001)
17. Raimond, Y., Sutton, C., Sandler, M.: Automatic interlinking of music datasets on the semantic web. In: First Workshop on Linked Data on the Web, Beijing, China (2008)
18. Stumme, G., Maedche, A.: FCA-Merge: Bottom-up merging of ontologies. In: International Joint Conference on Artificial Intelligence, Seattle, Washington, pp. 225–234 (2001)
19. Vatant, B., Wick, M.: Geonames ontology, `http://www.geonames.org/ontology/`