Chapter 1

Aligning Ontologies of Linked Data

Rahul Parundekar

University of Southern California

Craig A. Knoblock

University of Southern California

José Luis Ambite

University of Southern California

1.1 Introduction

Linked data is characterized by defining links between Semantic Web data resources using equivalence statements such as *owl:sameAs*, as well as other types of properties. Despite the increase in the number of linked instances in recent times, the absence of links at the concept level has resulted in heterogenous schemas, challenging the interoperability goal of the Semantic Web. For example, out of the 190 linked data sources surveyed in the latest cen-

sus[1], only 15 have mappings between their ontologies. The problem of schema linking, such as schema matching in databases and ontology alignment in the Semantic Web, has received much attention [2, 7, 3, 8]. Many approaches for linking schemas have been developed, including techniques that exploit linguistic, terminological, structural, or extensional properties of the sources.

In this chapter we present a novel extensional approach to generate alignments between ontologies of linked data sources. Similar to previous work on instance-based matching [6, 5, 10], we rely on linked instances to determine the alignments. Two concepts are equivalent if all (or most of) their respective instances are linked (by *owl:sameAs* or similar links). However, our search is not limited to the existing concepts in the ontology. We hypothesize new concepts by combining existing elements in the ontologies and seek alignments between these more general concepts. This ability to generalize allows our algorithm to find many more meaningful relationships between the ontologies.

The problem of finding alignments in ontologies between sources in linked data is non-trivial since one-to-one concept equivalences may not exist. In some sources the ontology is extremely rudimentary (e.g., *GeoNames* has only one class - *geonames:Feature*) and the alignment of such an impoverished ontology with a well developed one, such as *DBpedia*, is not particularly informative. In order to be successful in linking ontologies, we first need to generate more expressive concepts. The necessary information to do this is often present in the properties and values of the instances in the sources. For example, in *GeoNames* the values of the *featureCode* and *featureClass* properties provide useful information that can be used to find alignments with existing concepts in *DBpedia*, such as the alignment of the concept *geonames:featureClass=P* to *dbpedia:PopulatedPlace*. Therefore, our approach explores the space of concepts generated by value restrictions, which we will call *restriction classes* in the reminder of the paper. A *value restriction* is a concept constructor present in expressive description logics such as OWL2 DL ($\mathcal{SROIQ}$) [9]. We consider class assertions (*rdf:type*) and value restrictions on both object and data properties, which we will represent uniformly as $\{p = v\}$ and refer to as an *atomic restriction class*, where either p is an object property and v is a resource (including rdf:type=Class), or p is a data property and v is a literal. Associated with each *atomic restriction class* $\{p = v\}$ is a set of instances that extensionally defines the concept, where each instance has a value v asserted for its property p. We consider two *restriction classes* equivalent if their respective instance sets can be identified as equal after following the *owl:sameAs* (or similar) links. We also explore alignments between composite concepts, defined by conjunctions and disjunctions of *atomic restriction classes*.

We have developed algorithms to find alignments between atomic, conjunctive, and disjunctive *restriction classes* in linked data sources based on the extensions of the concepts (i.e. the sets of instances satisfying the definitions of the *restriction classes*). We believe that this is an important feature

[1] http://www4.wiwiss.fu-berlin.de/lodcloud/state/

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 readily be 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.

This chapter is organized as follows. First, we describe two linked data sources in the geospatial domain that we will use to explain our approach. Second, we present algorithms to generate three types of alignments: (i) equivalence and subset relations between *atomic restriction classes*, (ii) alignments between *conjunctive restriction classes* [13], and (iii) alignments between *restriction classes* formed using disjunctions (*concept coverings*) [14]. While doing so, we also describe how our approach is able to automatically curate existing linked data by identifying inconsistencies, incorrect values and possible linking errors. Third, we describe representative alignments discovered by our approach and present an evaluation of the results. Finally, we compare with related work, summarize our contributions, and discuss future work.

1.2 Linked Data Sources with Heterogeneous Ontologies

Linked data sources often conform to different, but related, ontologies that can be meaningfully linked [4, 12, 13, 14]. To illustrate our approach we use two sources with geospatial data, *GeoNames* & *DBpedia*, which have over 86,000 pairs of instances linked using the *owl:sameAs* property. It should be noted, however, that our algorithms are generic and can be used to align any two linked sources. *GeoNames* (`geonames.org`), contains about 7.8 million geographical features. Since its Semantic Web version was generated automatically from a simple relational database, it has a rudimentary ontology. All instances in *GeoNames* belong to a single class (*Feature*) with the type of the geographical data (e.g. mountains, lakes, etc.) encoded in the *featureClass* and *featureCode* properties. *DBpedia* (`dbpedia.org`), is a knowledge base that covers multiple domains and includes about 526,000 places and other geographical features. It uses a rich ontology with extensive concept hierarchies and relations to describe these instances.

1.3 Finding Alignments Across Ontologies

We find three types of alignments between the ontologies of linked data sources. First, we extract equivalent and subset alignments between *atomic*

restriction classes. These are the simplest alignments we define and are often interesting. We then use them as seed hypotheses to find alignments that are more descriptive. The second type of alignments we find are between *conjunctive restriction classes* in the two sources. The third type of alignments we find, Concept Coverings, are alignments where a larger concept from one source can be described with a union of smaller concepts from the other source.

1.3.1 Source Preprocessing

Before we begin exploring alignments, we perform some simple preprocessing 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 would not be useful. As an 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[2] has links to 237 articles in different languages. Each of these articles, however, could only be used to identify *Saudi Arabia*, so *restriction classes* based on *wikipediaArticle* would not yield useful concepts. Similarly, the latitude (*georss:lat*) and longitude (*georss:lat*) properties in *GeoNames* are also almost inverse functional properties and thus not useful concept constructors. On the other hand, the countryCode property in *GeoNames* has a *range* of 2-letter country codes that can be used to group instances into meaningful *restriction classes*.

1.3.2 Aligning Atomic Restriction Classes

Atomic *restriction classes* can be generated in each source automatically by exploring the space of distinct properties and their distinct values by the simple algorithm in Fig.1.1. Fig.1.2 illustrates the set comparison operations of our algorithm. We use two metrics P and R to measure the degree of overlap between *restriction classes*. In order to allow a certain margin of error induced by the data set, we use $P \geq \theta$ and $R \geq \theta$ (instead of $P = 1$ and $R = 1$, which would hold if there were no erroneous or missing links) in our score func-

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

tion. In our experiments we used a threshold $\theta = 0.9$, which was determined empirically, but can be changed as desired. For example, consider the alignment between *restriction classes* {*geonames:countryCode*=ES} from *GeoNames* and {*dbpedia:country* = *dbpedia:Spain*} from *DBpedia*. Based on the extension sets, our algorithm finds $|Img(r_1)| = 3198$, $|r_2| = 4143$, $|Img(r_1) \cap r_2| = 3917$, R =0.9997 and P = 0.9454. Thus, the algorithm considers this alignment as equivalent in an extensional sense. Our algorithm also finds that each of {*geonames:featureCode* = *S.SCH*} and {*geonames:featureCode* = *S.UNIV*} (i.e. Schools and Universities from *GeoNames*) are subsets of {*dbpedia:EducationalInstitution*}.

function ATOMICALIGNMENTS($Source_1$,$Source_2$)
  for all properties p_1 in $Source_1$, **all** distinct values $v_1 \in p_1$, **all** p_2 in $Source_2$, and **all** distinct $v_2 \in p_2$ **do**
    $r_1 \leftarrow \{p_1 = v_1\}$ // instances of $Source_1$ with $p_1 = v_1$
    $r_2 \leftarrow \{p_2 = v_2\}$
    $Img(r_1) \leftarrow$ instances of $Source_2$ linked to those in r_1
    $P \leftarrow \frac{|Img(r_1) \cap r_2|}{|r_2|}$, $R \leftarrow \frac{|Img(r_1) \cap r_b|}{|r_1|}$
    $alignment(r_1, r_2) \leftarrow$ [
    if $P \geq \theta$ *and* $R \geq \theta$ **then** $r_1 \equiv r_2$
    else if $P \geq \theta$ **then** $r_1 \subset r_2$
    else if $R \geq \theta$ **then** $r_2 \subset r_1$
    end if]
  end for
end function

FIGURE 1.1: Aligning *atomic restriction classes*

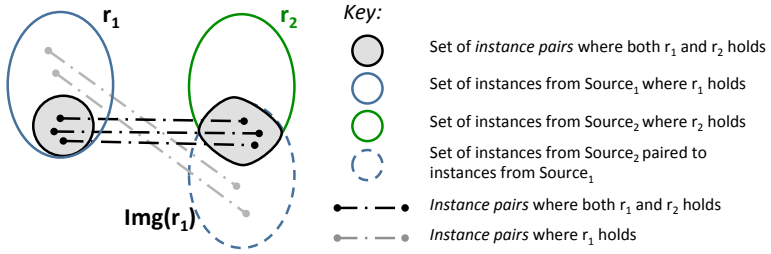

FIGURE 1.2: Comparing the linked instances from two ontologies

The complexity of ATOMICALIGNMENTS is $O(p^2m^2i\log(i))$, where p is the maximum number of distinct properties in the two sources, m is the maximum number of distinct values for any property, and i is number of instances in the largest *atomic restriction class*. Despite having a polynomial run-time, we also use certain optimization strategies for faster computation. For example, if we explore the properties lexicographically, the search space is reduced to

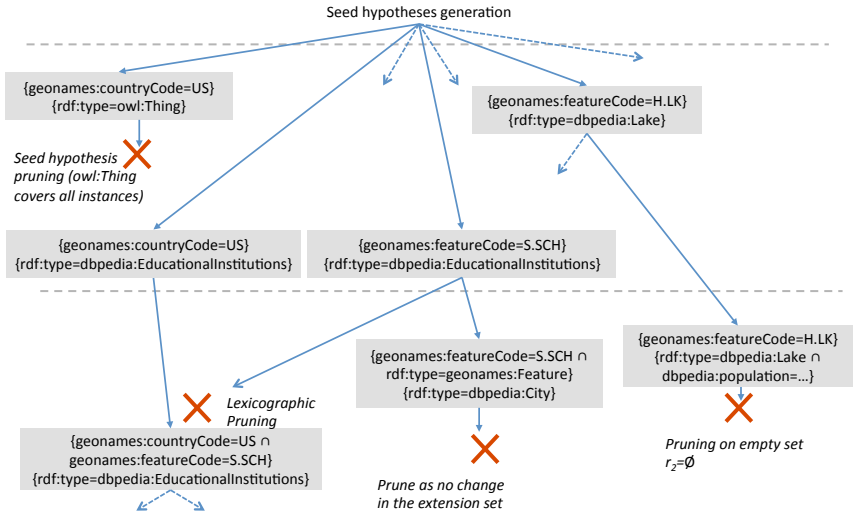

FIGURE 1.3: Exploring and pruning the space of alignments

half because of symmetry. Also, to qualify as an alignment hypothesis, the intersection of the *restriction classes* needs to have a minimum support, which we set experimentally to ten instances.

1.3.3 Aligning Conjunctive Restriction Classes

The second type of alignment we detect are those between *conjunctive restriction classes*. For example, the *conjunctive restriction class* 'Schools in the US', {*geonames:countryCode=US* ∩ *geonames:featureCode=S.SCH*}, is the intersection of the *atomic restriction classes* representing all schools in *GeoNames* and all features in the US.

We seed the search space with the alignments generated by ATOMICALIGNMENTS. Taking one hypothesis at a time, we can generate a new hypothesis from it by using the conjunction operator on one of the *atomic restriction classes* from the two sources to intersect it with another *atomic restriction class*. This process is shown in Fig.1.3. Since the space of alignment hypotheses is combinatorial, our algorithm exploits the set containment property of the hypotheses in a top-down fashion along with several systematic pruning features to manage the search space.

Pruning: Our algorithm prunes the search space in several ways. First, we prune those hypotheses where the number of supporting instance pairs is less than a given threshold. For example, the hypothesis [{*geonames:-featureCode=H.LK*}, {*rdf:type=dbpedia:Lake* ∩ *dbpedia:population=...*}] in Fig.1.3 is pruned since it has no support.

Second, if the number of instances of the new hypothesis formed after adding an *atomic restriction class* to one of the *restriction classes* did not change, then it means that adding the constraint did not specialize the current hypothesis. Any of its possible child hypotheses would also occur in some other branch of the search space. Because of this, we can prune this hypothesis. Fig.1.3 shows such pruning when the *atomic restriction class* {*rdf:type=geonames:Feature*} is added to the alignment [{*geonames:featureCode=S.SCH*}, {*rdf:type=dbpedia:City*}]. A special case of this pruning is when the seed hypothesis itself contains all instances in one of the sources. For example, the alignment [{*geonames:countryCode=US*}, {*rdf:type =owl:Thing*}].

Third, we prune hypotheses $[r'_1, r_2]$ where r'_1 is a refinement (subclass) of r_1 and $r_1 \cap r_2 = r_1$, as illustrated in Fig.1.4. In this case, imposing an additional restriction on r_1 to form r'_1 would not provide any immediate specialization. Any children $[r'_1, r'_2]$ of $[r'_1, r_2]$ that would make interesting alignments can be explored from the children of $[r_1, r'_2]$. We are choosing to prune half of the possible children of $[r_1, r_2]$, by skipping all $[r'_1, r_2]$ and investigating only $[r_1, r'_2]$ and its children. In practice, since we use $\theta = 0.9$, we ignore all children $[r'_1, r_2]$ when $|r_1| < |r_2|$. This still ensures that all possible hypotheses are explored. The same holds for the symmetrical case $r_1 \cap r_2 = r_2$.

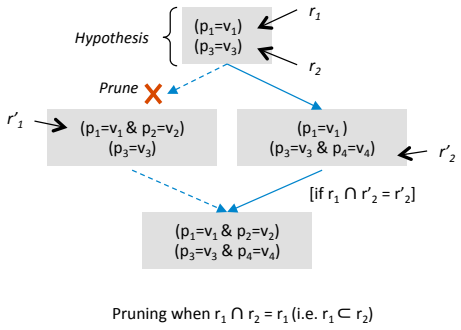

FIGURE 1.4: Pruning the hypotheses search space

Finally, to explore the space systematically, the algorithm specializes the restriction classes in a lexicographic order. For example, the addition of the restriction {*geonames:countryCode=US*} to [{*geonames:featureCode=S.SCH*}, {*rdf:type =dbpedia:EducationalInstitution*}] is pruned as shown in Fig.1.3. Also, as an optimization our algorithm only considers conjunctions of *atomic restriction classes* on different properties.

The algorithm to find *conjunctive restriction classes* is shown in Fig.1.5.

function CONJUNCTIVEALIGNMENTS($Source_1$,$Source_2$)
for all $[r_1, r_2] \in$ ATOMICALIGNMENTS($Source_1$,$Source_2$) **do** EXPLOREHYPOTHESES(r_1, r_2,$Source_1$,$Source_2$)
end for
end function
function EXPLOREHYPOTHESIS(r_1,r_2,$Source_a$,$Source_b$)
for all p_a in $Source_a$ occurring lexicographically after all the properties in r_1 and distinct v_a associated with p_a **do**
$r'_1 \leftarrow r_1 \cap \{p_a = v_a\}$
$alignment \leftarrow$ FINDALIGNMENT(r'_1,r_2)
if not SHOULDPRUNE(r'_1,r_2,$alignment$) **then**
$alignment(r'_1, r_2) \leftarrow alignment$
EXPLOREHYPOTHESES(r'_1, r_2)
end if
end for
for all p_b in $Source_b$ occurring lexicographically after all the properties in r_2 and distinct v_b associated with p_b **do**
$r'_2 \leftarrow r_2 \cap \{p_b = v_b\}$
$alignment \leftarrow$ FINDALIGNMENT(r_1,r'_2)
if not SHOULDPRUNE(r_1,r'_2,$alignment$) **then**
$alignment(r_1, r'_2) \leftarrow alignment$
EXPLOREHYPOTHESES(r_1, r'_2)
end if
end for
end function

FIGURE 1.5: Aligning *conjunctive restriction classes*

1.3.4 Eliminating Implied Alignments

From the resulting alignments of *conjunctive restriction classes* 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. Fig.1.6 explains the reduction process. 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., r'_1 is constructed by conjoining with an additional property-value pair to r_1). Fig.1.6 depicts 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. Dashed arrows denote implied containment relations.

A typical example of the reduction is Fig.1.6(e) where the result set contains a relation such that $r_1 \subset r_2$ and $r'_1 \subset r_2$. Since $r'_1 \subset r_1$, 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

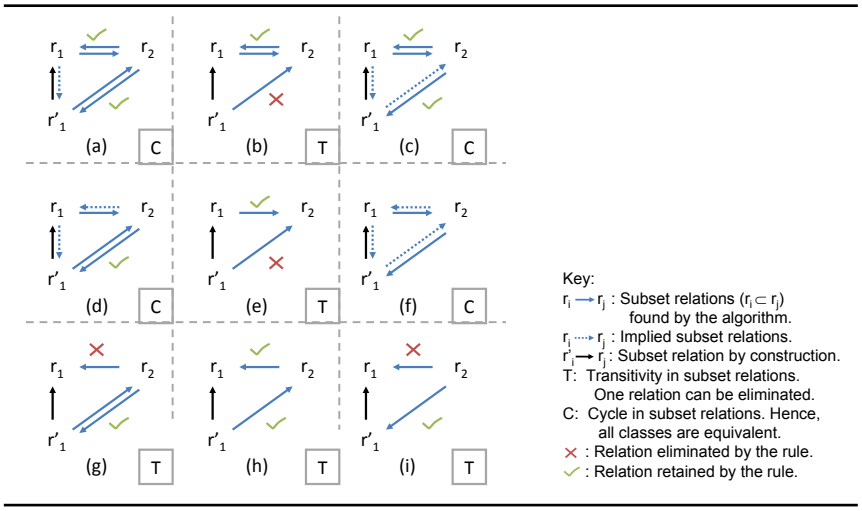

FIGURE 1.6: Eliminating Implied Alignments

hence remove $r'_1 \subset r_2$. Other such transitive relations and their reductions are depicted with a 'T' in the bottom-right corner of each cell.

Another case can be seen in Fig.1.6(d) where the subsumption relationships found in the alignment results can only hold if all 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. Other similar cases can be seen in Fig.1.6(a), (c) and (f) where the box on the bottom-right is has a 'C' (cycle). 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 Fig.1.6(a), (c), (d), (f) to identify equivalences until quiescence. Then it applies the 'T' rules to eliminate hypotheses that are not needed.

With our reduced thresholds, the removal of implied alignments may be sometimes difficult. For example, we may detect both alignments [{*geonames:featureCode=H.LK*} = {*rdf:type=dbpedia:BodyOfWater*}] and [{*geonames:featureCode=H.LK*} = {*rdf:type=dbpedia:Lake*}] since the number of lakes might be substantially larger than other bodies of water[3], and passing our threshold of 0.9. In such a case, we choose the alignment that is a "better fit". To do this, we look at P and R values of both the alignments

[3]Though this condition may not represent real world phenomena, it can be a frequent occurrence in aligning the data from any two linked data sources, since the evidence we use is only of the linked instances

and select the alignment that has the higher F-measure (harmonic mean) of the two.

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 (RDF-S/OWL) provided by the sources for the subclass relationships.

1.3.5 Finding Concept Coverings

The CONJUNCTIVEALIGNMENTS algorithm may produce a very large number of subset relations, even after the reduction algorithm. Analyzing the results of aligning *DBpedia* and *GeoNames* in [13], we noticed that these subset alignments follow common patterns. For example, we found that Schools from *GeoNames* were subsets of Educational Institutions from *DBpedia*. Similarly, Universities from *GeoNames* were also subsets of Educational Institutions. Though each of these alignments taken individually were only slightly informative, we realized that if taken in combination, we could find much more significant equivalence alignments. For example, we found that the concept Education Institution in *DBpedia* covers Schools, Colleges, and Educational institutions from *GeoNames* completely. With this motivation, we developed an approach for finding *concept coverings*.

In order to find *concept coverings*, we use the subclasses and equivalent alignments found with *atomic restriction classes* to try and align a larger concept from one ontology with a union of smaller subsumed concepts in the other ontology. To define a larger concept, we group its subclasses from the other source that have a common property and check whether they cover the larger concept. By keeping the larger *restriction class* atomic and by grouping the smaller *restriction classes* with a common property, we are able to find intuitive definitions while keeping the problem tractable. The disjunction operator that groups the smaller *restriction classes* is defined such that *i)* the concept formed by the disjunction of the classes represents the union of their set of instances, *ii)* the property for all the smaller aggregated *atomic restriction classes* is the same. We then try to detect the alignment between the larger concept and the union *restriction class* by using an extensional approach similar to the previous step. The algorithm for generating the hypotheses and the alignments is shown in Fig.1.7.

Since all smaller classes are subsets of the larger *restriction class*, $P_U \geq \theta$ holds by construction. We used $\theta = 0.9$ in our experiments to determine subset relation in the other direction. The smaller *restriction classes* that were omitted in the first step (ATOMICALIGNMENTS because of insufficient support size of their intersections (e.g., $\{$*geonames:featureCode* = *S.SCHC*$\}$), were included in constructing U_S for completeness.

function CONCEPTCOVERINGS($Source_1$,$Source_2$)
for all alignments $[U_L, r_2] \in$ ATOMICALIGNMENTS($Source_1$,$Source_2$), with larger concept $U_L = \{p_L = v_L\}$ from $Source_1$ and multiple classes $r_2 = \{p_S = v_i\}$ from $Source_2$ that can be partitioned on property p_S **do**
for all smaller concepts $\{p_S = v_i\}$ **do**
$U_S \leftarrow \{p_S = \{v_1, v_2, ...\}\}$ // union *restriction class*
$U_A \leftarrow Img(U_L) \cap U_S$, $P_U \leftarrow \frac{|U_A|}{|U_S|}$, $R_U \leftarrow \frac{|U_A|}{|U_L|}$
if $R_U \geq \theta$ **then** $alignment(r_1, r_2) \leftarrow U_L \equiv U_S$
end if
end for
end for
end function

FIGURE 1.7: Finding Concept Coverings

Fig.1.8 provides an example of the approach. The first step is able to detect that alignments such as $\{geonames{:}featureCode = S.SCH\}$, $\{geonames{:}featureCode = S.SCHC\}$, $\{geonames{:}featureCode = S.UNIV\}$ are subsets of $\{rdf{:}type = dbpedia{:}EducationalInstitution\}$. As can be seen in the Venn diagram in Fig.1.8, U_L is $Img(\{rdf{:}type = dbpedia{:}EducationalInstitution\})$, U_S is $\{geonames{:}featureCode = S.SCH\} \cup \{geonames{:}featureCode = S.SCHC\} \cup \{geonames{:}featureCode = S.UNIV\}$, and U_A is the intersection of the two. Upon calculation we find that R_U for the alignment of *dbpedia:EducationalInstitution* to {*geonames:featureCode*= {S.SCH, S.SCHC, S.UNIV}} is 0.98 (greater than θ). We can thus confirm the hypothesis and consider U_L and U_S as equivalent. The experiments in Section 1.4 describe additional examples of *concept coverings*.

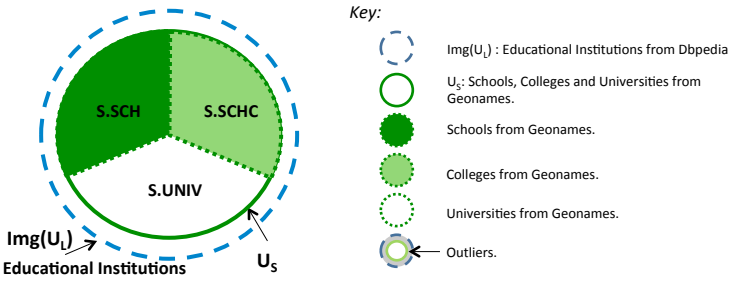

FIGURE 1.8: Concept covering of Educational Institutions from *DBpedia*

1.3.6 Curating Linked Data

It turns out that the outliers, the instances of the *restriction classes* that do not satisfy subset relations despite the error margins, are often due to

incorrect and missing links or assertions. Our algorithm is also able to detect these outliers, thus providing a novel method to curate existing linked data.

For example, ATOMICALIGNMENTS correctly aligned the country Spain in *DBpedia* and *GeoNames*: {*dbpedia:country = Spain*} ≡ {*geonames:-countryCode*} = {*ES*}}. However, one outlier instance of {*dbpedia:country = Spain*} had the country code IT (Italy) in *GeoNames*, suggesting an incorrect link/assertion. The algorithm flagged this situation as a possible error since there is overwhelming support for 'ES' being the country code of Spain. As another example, CONCEPTCOVERINGS aligned {*rdf:type = dbpedia:EducationalInstitution*} to {*geonames:featureCode*} = {*S.SCH, S.SCHC, S.UNIV*}} and identified 8 outliers (cf. alignment #12 in Table 1.4) . For {*rdf:type = dbpedia:EducationalInstitution*}, 396 instances out of the 404 Educational Institutions were accounted for as having their *geonames:featureCode* as one of *S.SCH, S.SCHC or S.UNIV*. From the 8 outliers, 1 does not have a *geonames:featureCode* property asserted. The other 7 have their feature codes as either S.BLDG (3 buildings), S.EST (1 establishment), S.HSP (1 hospital), S.LIBR (1 library) or S.MUS (1 museum). This case requires more sophisticated curation and the outliers may indicate a case for multiple inheritance. For example, the hospital instance in geonames may be a medical college that could be classified as a university. Other examples appear in Section 1.4.

In summary our alignment algorithms provide a powerful tool to quickly focus on links that require human curation or that could be automatically flagged as problematic, and it provides evidence for the errors.

1.4 Results

The results of the three algorithms for aligning *GeoNames* and *DBpedia* are shown below in Table 1.1. In all, we were able to detect about 580 (263 + 45 + 221 + 51) equivalent alignments including both atomic and complex *restriction classes*, along with 15,376 (4,946 + 5,494 + 4,400 + 536) subset relations.

1.4.1 Representative Examples of Atomic Alignments

Table 1.2 shows some examples of the alignments that we were able to detect between *atomic restriction classes*. In the table, column 2 shows the *atomic restriction class* from *GeoNames* and column 3 shows the *atomic restriction class* from *DBpedia*. The relationship detected between the two *atomic restriction classes* is shown in column 4, while the P and R scores used for detecting the relation are shown in columns 5 & 6. Column 7 defines the size of the intersection set of these two classes. Since we consider the use of the *rdf:type* property to form an *atomic restriction class* as a valid constructor,

TABLE 1.1: Alignments found between *GeoNames* and *DBpedia*

GeoNames* &*DBpedia* *atomic restriction classes	**#Alignments**
Equivalent Alignments	263
Subset Alignments with larger class from *GeoNames*	4,946
Subset Alignments with larger class from *DBpedia*	5,494
conjunctive restriction classes	
Equivalent Alignments	45
Subset Alignments with larger class from *GeoNames*	4,400
Subset Alignments with larger class from *DBpedia*	536
concept coverings	
Concept Coverings with larger class from *GeoNames*	221
Concept Coverings with larger class from *DBpedia*	51

we are able to find alignments with traditional concepts in the ontology. For example, in alignment #1, we can see that the concept for PopulatedPlace in *DBpedia* is equal to the *atomic restriction class* depicting the set of instances in *GeoNames* where the value of the Feature Class is *geonames:P*. Similarly, the concept of things with a Feature Class of *geonames:H* in *GeoNames* is equivalent to the *BodyOfWater* concept in *DBpedia*.

Our *atomic restriction class* constructors also allow us to detect more interesting alignments. For example, we correctly identify the equality relation between the concept denoting the country Spain in both sources, formed by {*geonames:countryCode=ES*} in *GeoNames* and {*dbpedia:country=dbpedia:Spain*} in *DBpedia* (alignment #3). Similarly, we can align various geographical regions, as shown by the alignment #4 of the *atomic restriction classes* denoting the administrative division of Sicily in either source. Since the alignments that our algorithm generates capture the actual relationship between the data in the two sources rather than what an ontology disconnected from the data would assert, we are able to find interesting patterns of ontology mismatch as shown in alignment #5. Even though one would expect that the concept of Mountains is the same in *GeoNames* and *DBpedia*, in reality, the class of mountains in *GeoNames* is a subset of mountains in *DBpedia*. Upon inspection we found this to be because the concept in *GeoNames* did not include hills (T.HLL), peaks (T.PK), some volcanos (T.VLC), etc., which were part of the definition of mountains in *DBpedia*.

In some cases, our algorithm produced incorrect results because of our relaxed threshold assumption. For example, the algorithm incorrectly asserted that Schools in *GeoNames* are equivalent to Educational Institutions in *DBpedia*(alignment #6), while they are in fact a subset. Upon inspection of other alignments in the sources (e.g., alignment #7, which shows that Universities in *GeoNames* are Educational Institutions in *DBpedia*), we decided to rectify

this by exploring alignments generated from more complex *restriction classes*, i.e., alignments between *conjunctive restriction classes* and *concept coverings*.

1.4.2 Representative Examples of Conjunctive Alignments

Table 1.3 shows the alignments CONJUNCTIVEALIGNMENTS found between *conjunctive restriction classes* in *GeoNames* and *DBpedia*. Alignments #8 and #9 follow from the results already discovered in Table 1.2. Alignment #8, 'Populated Places in the US', and alignment #9, 'Body of Water in New Zealand', are refinements of alignments #1 and #2 respectively.

We also found other interesting alignments where the aligned concepts have properties that are related, e.g., the relation between some states in the US and their time zones. This can be seen in alignment #10, where we detected that Settlements in the state of Louisiana (in *GeoNames*) belonged to the North American Central Time Zone. Another example is the assignment of area codes for telephone numbers in the US based on geographic divisions. This is illustrated by alignment #11, where we identified that places in the state of North Dakota used the area code of 701. In some alignments that we found, our algorithm was able to generate results that showed the skew in ontological concepts generated by the specialization. In particular, because of the less data available for highly specialized classes, some alignments demonstrated that concepts can change in their meaning. For example, alignment #12 shows that places in Senegal with Feature Class 'P' in *GeoNames* are aligned with Towns (as opposed to *dbpedia:PopulatedPlace*s from *DBpedia*).

1.4.3 Representative Examples of Concept Coverings

Some representative examples of the *concept coverings* found are shown in Table 1.4. In the table, for each *concept covering*, column 2 describes the large *restriction class* from $Source_1$ and column 3 describes the union of the (smaller) classes on $Source_2$ with the corresponding property and value set. The score of the covering is noted in column 4 ($R_U = \frac{|U_A|}{|U_L|}$) followed by $|U_A|$ and $|U_L|$ in columns 5 and 6. Column 7 shows the outliers, i.e. values v_2 of property p_2 that form *restriction classes* that are not direct subsets of the larger *restriction class*. Each of these outliers also has a fraction with the number of instances that belong to the intersection over the the number of instances of the smaller *restriction class* (or $\frac{|Img(r_1) \cap r_2|}{|r_2|}$). One can see that the fraction is less than our relaxed subset score. If the value of this fraction was greater than the relaxed subset score (i.e. $\theta = 0.9$), the set would have been included in column 3 instead. For example, the *concept covering* #13 of Table 1.4 is the Educational Institution example described earlier. It shows how educational institutions from *DBpedia* are equivalent to the union of schools, colleges and universities in *GeoNames*. Column 4, 5 and 6 explain the alignment score R_U (0.98, or 98%), the size U_A (396) and the size of

U_L (404). The outliers (S.BLDG, S.EST, S.LIBR, S.MUS, S.HSP) along with their P fractions appear in column 7. Thus, 403 of the total 404 instances were identified as either part of the covering or the outliers. The remaining instance did not have a *geonames:featureCode* property asserted.

A common pattern of *concept coverings* discovered was the alignments between administrative divisions at different levels in the geospatial sources. For example, alignment #14 shows the sub-divisions of *Basse-Normandie*. Table 1.4 shows and explains additional examples of *concept coverings*. The complete set of alignments discovered by our algorithm is available online.[4]

1.4.4 Outliers

Our algorithms identify two main types of inconsistencies: **(i)** *Incorrect instance alignments* - outliers arising out of a possible erroneous equivalence link between instances (e.g., in alignment #15, a hill is linked to an airport, etc.), and **(ii)** *Incorrect values for properties* - outliers arising out of possible erroneous assertion for a property (e.g., in alignments #17 and #18, flags of countries appear as values for the *country* property).

Our *concept covering* algorithm was able to detect outliers in alignments of *atomic restriction classes*. Alignment #16 shows the outliers detected for alignment #3 of Table 1.2 (i.e. the alignment of the country Spain, where the only outlier had its country as Italy). In the alignments in the table, we also mention the classes that these inconsistencies belong to along with their support. As was the case with detecting the alignments, we were unable to detect some outliers if there was insufficient support for coverage due to missing instances or missing links.

1.4.5 Precision and Recall of Country Alignments

Since manually establishing ground truth for all possible concept coverings in *GeoNames* and *DBpedia* is infeasible, we selected a representative class for which we could compute the exact alignments, namely the set of countries. These alignments follow a common pattern, with *dbpedia:country* properties aligning to *geonames:countryCode* properties. A ground truth was established by manually checking what possible country alignments were present in the two sources. Even then, establishing the ground truth needed some insight. For example, Scotland, England, Wales, Northern Ireland, and the United Kingdom are all marked as countries in *DBpedia*, while in *GeoNames*, the only corresponding country is the United Kingdom. In cases like these, we decided to relax the evaluation constraint of having an alignment with a country from either of these, as correct. Another similar difficulty was in cases where militarily occupied territories were marked as countries (e.g., the Golan Heights region occupied by Israel is marked as a *dbpedia:country*).

[4] http://www.isi.edu/integration/data/UnionAlignments

Out of the 63 country alignments detected, 26 were correct. There were 27 alignments that had a '.svg' file appearing as value of the country property in *DBpedia*. We would have detected such *concept coverings*, had such assertions for the country property been correct. Since this is a problem with the data and not our algorithm, we consider these 27 as correct for this particular evaluation. We thus get a precision of 84.13% ((26+27) out of 63). The two sources contained around 169 possible country alignments between them, including countries with a '.svg' value for the country property. There were many alignments in the ground truth that were not found because the system did not have enough support ($R < 0.9$) to pass our threshold. Accordingly, the recall was 31.36%, for an F1-measure of 45.69%.

1.4.6 Representative Alignments from Other Domains

Our algorithms are generic and can find alignments between any two sources with linked instances. Table 1.5 shows the alignments that our approach finds in the Biological Classification and Genetics domains. In particular, the table shows the alignments between classes from the animal and plant kingdoms in *Geospecies* & *DBpedia* and between classes from the *MGI* & *GeneID* databases in `bio2rdf.org`. A detailed description of these sources and corresponding alignment results appears in Parundekar et al. [14].

1.5 Related Work

Ontology alignment and schema matching has received much attention over the years [2, 7, 3, 8] with a renewed interest recently due to the rise of the semantic web. In linked data, even though most work done is on linking instances across different sources, an increasing number of authors have looked into aligning the ontologies of linked data sources. BLOOMS [11] uses a central forest of concepts derived from topics in Wikipedia. This approach fails to find alignments with *GeoNames* because of its rudimentary ontology (single *Feature* class). Its successor, BLOOMS+ [12], aligns ontologies of linked data with an upper-level ontology called Proton using contextual information. BLOOMS+ is marginally more successful than its predecessor in finding alignments between *GeoNames* & Proton and *DBpedia* & Proton (precision = 0.5% & 90% respectively). *AgreementMaker* [4] is a dynamic ontology mapping approach that uses similarity measures along with a mediator ontology to find mappings using the labels of the classes. From the subset and equivalent alignments between *GeoNames* (10 concepts) and *DBpedia* (257 concepts), it achieves a precision of 26% and a recall of 68%. In comparison, for *GeoNames* and *DBpedia*, we achieve a precision of 64.4%. But this comparison does not reflect that we find concept coverings in addition to one-to-one alignments,

TABLE 1.2: Example Alignments of *atomic restriction classes* from *GeoNames* and *DBpedia*

#	***GeoNames***	***DBpedia***	**Rel**	***P***	***R***	$\lvert I(r_1) \cap r_2 \rvert$
1	geonames:featureClass=geonames:P	rdf:type=dbpedia:PopulatedPlace	=	99.57	90.47	70658
2	geonames:featureClass=geonames:H	rdf:type=dbpedia:BodyOfWater	=	91.11	99.08	1939
3	geonames:countryCode=ES	dbpedia:country=dbpedia:Spain	=	94.54	99.97	3917
4	geonames:parentADM1= http://sws.geonames.org/2523119/	dbpedia:region= dbpedia:Sicily	=	92.13	1	328
5	geonames:featureCode=geonames:T.MT	rdf:type=dbpedia:Mountain	$\subset$	96.8	78.4	1721
6	geonames:featureCode=geonames:S.SCH	rdf:type=dbpedia:EducationalInstitution	=	93.31	93.31	377
7	geonames:featureCode=geonames:S.UNIV	rdf:type=dbpedia:EducationalInstitution	$\subset$	100	3.71	15

TABLE 1.3: Example Alignments of *conjunctive restriction classes* from *GeoNames* and *DBpedia*

#	***GeoNames***	***DBpedia***	**Rel**	***P***	***R***	$\lvert I(r_1) \cap r_2 \rvert$
8	geonames:featureClass=geonames:P & geonames:countryCode=US	rdf:type=dbpedia:PopulatedPlace & dbpedia:country=dbpedia:United_States	=	97.15	96.72	26061
9	geonames:countryCode=NZ & geonames:featureClass=H	rdf:type=dbpedia:BodyOfWater & dbpedia:country=dbpedia:New_Zealand	=	92.59	100	50
10	geonames:featureClass=geonames:P & geonames:parentADM1= http://sws.geonames.org/4331987/	dbpedia:daylightSavingTimeZone= dbpedia:North_American_Central_Time_Zone	=	97.41	99	1091
11	geonames:featureClass=geonames:P & geonames:parentADM1= http://sws.geonames.org/5690763/	dbpedia:areaCode=701@en	=	98.09	96.52	361
12	geonames:featureClass=geonames:P & geonames:countryCode=SN	dbpedia:type=Town & dbpedia:country=dbpedia:Senegal	=	92.59	100	25

TABLE 1.4: Example *concept coverings* from *GeoNames* and *DBpedia*

#	r_1	$p_2 = \{v_2\}$	$R_U = \frac{\lvert U_A \rvert}{\lvert U_L \rvert}$	$\lvert U_A \rvert$	$\lvert U_L \rvert$	***Outliers***
***DBpedia* (larger) - *GeoNames* (smaller)**						
13	As described in Section 1.4, Schools, Colleges and Universities in *GeoNames* make Educational Institutions in *DBpedia*					
	rdf:type = dbpedia:EducationalInstitution	*geonames:featureCode =* {S.SCH, S.SCHC, S.UNIV}	98.01	396	404	S.BLDG (3/122), S.EST (1/13), S.LIBR (1/7), S.HSP (1/31), S.MUS (1/43)
14	We confirm the hierarchical nature of administrative divisions with alignments between administrative units at two different levels.					
	dbpedia:region = dbpedia:Basse-Normandie	*geonames:parentADM2 =* {geonames:2989247, geonames:2996268, geonames:3029094}	100.0	754	754	
15	In aligning airports, an airfield should have been an an airport. However, there was not enough instance support.					
	rdf:type = dbpedia:Airport	*geonames:featureCode =* {S.AIRB, S.AIRP}	99.24	1981	1996	S.AIRF (9/22), S.FRMT (1/5), S.SCH (1/404), S.STNB (2/5) S.STNM (1/36), T.HLL (1/61)
16	The concepts for the country Spain are equal in both sources. The only outlier has its country as Italy, an erroneous link.					
	dbpedia:country = dbpedia:Spain	*geonames:countryCode =* {*ES*}	0.9997	3917	3918	IT (1/7635)
***GeoNames* (larger) - *DBpedia* (smaller)**						
17	The Alignment for Netherlands should have been as straightforward as #2. However we have possible alias names, such as *The Netherlands* and *Kingdom of Netherlands*, as well a possible linkage error to *Flag_of_the_Netherlands.svg*					
	geonames:countryCode = NL	*dbpedia:country =* {dbpedia:The_Netherlands, dbpedia:Flag_of_the _Netherlands.svg, dbpedia:Netherlands}	98.02	1939	1978	dbpedia:Kingdom_of _the_Netherlands (1/3)
18	The error pattern in #5 seems to repeat systematically, as can be seen from this alignment for the country of Jordan.					
	geonames:countryCode = JO	*dbpedia:country =* {dbpedia:Jordan, dbpedia:Flag_of_Jordan.svg}	95.0	19	20	

TABLE 1.5: Example alignments from *Geospecies-DBpedia* and *GeneID-MGI*

#	r_1	$p_2 = \{v_2\}$	$R_U = \frac{\|U_A\|}{\|U_L\|}$	$\|U_A\|$	$\|U_L\|$	***Outliers***
***DBpedia* (larger) - *Geospecies* (smaller)**						
19	Species from *Geospecies* with the order names Anura, Caudata & Gymnophionia are all Amphibians We also find inconsistencies due to misaligned instances, e.g. one amphibian was classified as a Turtle (Testudine) .					
	rdf:type = dbpedia:Amphibian	*geospecies:hasOrderName =* {Anura, Caudata, Gymnophionia}	99.0	90	91	Testudines (1/7)
20	Upon further inspection of #18, we find that the culprit is a Salamander					
	rdf:type = dbpedia:Salamander	*geospecies:hasOrderName =* {Caudata}	94.0	16	17	Testudines (1/7)
***Geospecies* (larger) - *DBpedia* (smaller)**						
21	We can detect that species with order Chiroptera correctly belong to the order of Bats. Unfortunately, due to values of the property being the literal "Chiropta@en", the alignment is not clean.					
	geospecies:hasOrderName = Chiroptera	*dbpedia:ordo =* {Chiroptera@en, dbpedia:Bat}	100.0	111	111	
***GeneID* (larger) - *MGI* (smaller)**						
22	The classes for Pseudogenes align.					
	bio2rdf:subType = pseudo	*bio2rdf:subType =* {Pseudogene}	93.0	5919	6317	Gene (318/24692)
23	The Mus Musculus (house mouse) genome is composed of complex clusters, DNA segments, Genes and Pseudogenes.					
	bio2rdf:xTaxon = taxon:10090	*bio2rdf:subType =* {Complex Cluster/Region, DNA Segment, Gene, Pseudogene}	100.0	30993	30993	
***MGI* (larger) - *GeneID* (smaller)**						
24	We find alignments like #23 , which align the gene start (with the chromosome) in *MGI* with the location in *GeneID*. As can be seen, the values of the locations (distances in centimorgans) in *GeneID* contain the chromosome as a prefix. Inconsistencies are also seen, e.g. in #23, the value is wrongly assigned "" or '5'.					
	mgi:genomeStart = 1	*geneid:location =* {1, 1 0.0 cM, 1 1.0 cM, 1 10.4 cM, ...}	98.0	1697	1735	""(37/1048) 5 (1/52)

which other approaches do not. We are able to find such alignments because of our use of *restriction classes*, as in the case of aligning the rudimentary ontology of *GeoNames* with *DBpedia*. We believe that since other approaches do not use such constructs to generate refined concepts, they would fail to find alignments like the Educational Institutions example (alignment #1).

Extensional techniques and concept coverings have also been studied in the past [10]. Völker et al. [16] describe an extensional approach that uses statistical methods for finding alignments. This work induces schemas for RDF data sources by generating OWL-2 axioms using an intermediate associativity table of instances and concepts and mining associativity rules from it. The GLUE [5] system is a instance-based matching algorithm, which first predicts the concept in the other source that instances belong to using machine learning. GLUE then hypothesizes alignments based on the probability distributions obtained from the classifications. Our approach, in contrast, depends on the existing links (in linked data), and hence reflects the nature of the source alignments. *CS*R [15] is a similar work to ours that tries to align a concept from one ontology to a union of concepts from another ontology. It uses the similarity of properties as features in predicting the subsumption relationships. It differs from our approach in that it uses a statistical machine learning approach for detection of subsets rather than the extensional approach. Atencia et al. [1] provide a formalization of weighted ontology mappings that is applicable to extensional matchers like ours.

1.6 Conclusion

We described an approach to identifying alignments between atomic, conjunctive and disjunctive *restriction classes* in linked data sources. Our approach produces alignments where concepts at different levels in the ontologies of two sources can be mapped even when there is no direct equivalence or only rudimentary ontologies exist. Our algorithm is also able to find outliers that help identify erroneous links or inconsistencies in the linked instances.

In future work, we want to find more complete descriptions for the sources. Our preliminary findings show that our results can be used to identify patterns in the properties. For example, the *countryCode* property in *GeoNames* is closely associated with the *country* property in *DBpedia*, though their ranges are not exactly equal. By mining rules from the generated alignments, we will be closer to the interoperability vision of the Semantic Web. We also intend to use the outliers to feed the corrections back to the sources, particularly *DBpedia*, and to the RDF data quality watchdog group `pedantic-web.org`. To achieve this satisfactorily, we not only need to point out the instances that have errors, but suggest why those errors occurred, that is, whether they were due to incorrect assertions or missing links.

Bibliography

[1] Manuel Atencia, Alexander Borgida, Jérôme Euzenat, Chiara Ghidini, and Luciano Serafini. A formal semantics for weighted ontology mappings. *The Semantic Web–ISWC 2012*, pages 17–33, 2012.

[2] Zohra Bellahsene, Angela Bonifati, and Erhard Rahm. *Schema Matching and Mapping*. Springer, 1st edition, 2011.

[3] P.A. Bernstein, J. Madhavan, and E. Rahm. Generic schema matching, ten years later. *Proceedings of the VLDB Endowment*, 4(11), 2011.

[4] I.F. Cruz, M. Palmonari, F. Caimi, and C. Stroe. Towards on the go matching of linked open data ontologies. In *Workshop on Discovering Meaning On The Go in Large Heterogeneous Data*, page 37, 2011.

[5] A.H. Doan, J. Madhavan, P. Domingos, and A. Halevy. Ontology matching: A machine learning approach. *Handbook on ontologies*, pages 385–404, 2004.

[6] M. Duckham and M. Worboys. An algebraic approach to automated geospatial information fusion. *International Journal of Geographical Information Science*, 19(5):537–558, 2005.

[7] Jerome Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, 2007.

[8] Avigdor Gal. *Uncertain Schema Matching*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[9] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The even more irresistible SROIQ. In *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 57–67, Lake District of the United Kingdom, June 2006.

[10] A. Isaac, L. Van Der Meij, S. Schlobach, and S. Wang. An empirical study of instance-based ontology matching. *The Semantic Web–ISWC 2007*, pages 253–266, 2007.

[11] P. Jain, P. Hitzler, A. Sheth, K. Verma, and P. Yeh. Ontology alignment for linked open data. *The Semantic Web–ISWC 2010*, pages 402–417, 2010.

[12] P. Jain, P. Yeh, K. Verma, R. Vasquez, M. Damova, P. Hitzler, and A. Sheth. Contextual ontology alignment of LOD with an upper ontology: A case study with proton. *The Semantic Web: Research and Applications–ESWC 2011*, pages 80–92, 2011.

[13] Rahul Parundekar, Craig A. Knoblock, and José Luis Ambite. Linking and building ontologies of linked data. *The Semantic Web–ISWC 2010*, 2010.

[14] Rahul Parundekar, Craig A. Knoblock, and José Luis Ambite. Discovering concept coverings in ontologies of linked data sources. *The Semantic Web–ISWC 2012*, pages 427–443, 2012.

[15] V. Spiliopoulos, A. Valarakos, and G. Vouros. CSR: discovering subsumption relations for the alignment of ontologies. *The Semantic Web: Research and Applications–ESWC 2008*, pages 418–431, 2008.

[16] J. Völker and M. Niepert. Statistical schema induction. *The Semantic Web: Research and Applications–ESWC 2011*, pages 124–138, 2011.