Bayes-ReCCE, a Bayesian model for detecting Restriction Class Correspondences in Linked Open Data knowledge bases

Brian Walshe, Rob Brennan, Declan O'Sullivan

ADAPT Centre for Digital Content Technology

Knowledge and Data Engineering Group,
School of Computer Science and Statistics,
Trinity College Dublin
Ireland

Abstract: Linked Open Data consists of a large set of structured data knowledge bases which have been linked together, typically using equivalence statements. These equivalences usually take the form of owl:sameAs statements linking individuals, but links between classes are far less common Often, the lack of linking between classes is because the relationships cannot be described as elementary one to one equivalences. Instead, complex correspondences referencing multiple entities in logical combinations are often necessary if we want to describe how the classes in one ontology are related to classes in a second ontology. In this paper we introduce a novel Bayesian Restriction Class Correspondence Estimation (Bayes-ReCCE) algorithm, an extensional approach to detecting complex correspondences between classes. Bayes-ReCCE operates by analysing features of matched individuals in the knowledge bases, and uses Bayesian inference to search for complex correspondences between the classes these individuals belong to. Bayes-ReCCE is designed to be capable of providing meaningful results even when only small amounts of matched instances are available. We demonstrate this capability empirically, showing that the complex correspondences generated by Bayes-ReCCE have a median F1 score of over 0.75 when compared against a gold standard set of complex correspondences between Linked Open Data knowledge bases covering the geographical and cinema domains. In addition we discuss how metadata produced by Bayes-ReCCE can be included in the correspondences to encourage reuse by allowing users to make more informed decisions on the meaning of the relationship described in the correspondences.

1 Introduction

Linked Open Data provides access to a wealth of information in standardised and navigable form, designed to enable these data to be combined easily. Bizer et al. [1] note however that "… most Linked Data applications display data from different sources alongside each other but do little to integrate it further. To do so does require mapping of terms from different vocabularies

to the applications target schema". Links usually take the form of owl:sameAs statements linking individuals, but links between classes are far less common [25]. Heterogeneity issues, such as differences in class scope or hierarchy granularity however mean that simple one to one correspondences between atomic classes are not always enough to describe the mappings between schemas, or more generally, ontologies. The YAGO2 [2] knowledge base, for example, contains a rich class hierarchy based on WordNet [3], and includes many professions described as classes. An instance of a person in YAGO2 who is a film director, belongs to the class *yago:FilmDirector*. In contrast, version 3.9 DBpedia [4] has a shallower class hierarchy, with professions described as attribute-values, not classes. In this version of the DBpedia ontology there is no named class for film directors. If one to one mappings between named classes is the only mechanism available, then we could say that *yago:FilmDirector* maps to *dbpedia:Person* with a subsumption relationship; but this does not describe which members of the class Person are film directors. If, instead, complex correspondences between non-atomic classes can be used, then it could be asserted that *yago:FilmDirector* corresponds with the set of instances of Person in DBpedia with the attribute *dbpedia-owl:occupation* set to *dbpedia:Film_director*. More formally, correspondences where at least one of the entities described in the correspondence is non-atomic are known as *complex correspondences* [5].

Research has shown that complex correspondences can be classified into commonly reoccurring Correspondence Patterns [6]. Extensional methods, which compare the instance sets of classes using some metric such as the Jaccard index, have been shown to be capable of detecting complex correspondences between ontologies used in Linked Open Data [7][8]. However, extensional based approaches have several issues. When only small amounts of instance data are available they can give high scores to spurious matches, and when the amount of data are large, the search space of potential correspondences can grow very quickly. A more subtle problem, which we will show in section 3, is that directly comparing the instance sets of two classes to test similarity is not consistent with the Open World Assumption. Furthermore with existing extensional approaches there is an a priori assumption that all forms of complex correspondences are equally probable, and the approaches do not provide a systematic way for us to specify any prior beliefs we have that certain patterns of correspondences may be more probable than others.

In this article we propose Bayes-ReCCE, a scalable complex correspondence detection algorithm which uses Bayesian statistics to estimate the true Jaccard index of the classes being compared, and which provides a method to specify prior beliefs about certain patterns of correspondence being more or less probable than others. Bayes-ReECCE presents the most probable correspondences to a user, combined with a summary of the evidence for each of these correspondences. Using the probability measure for the correspondence and examining the evidence allows a user to make a more informed decision on whether to accept or reject the correspondence.

We make the following contributions:

- A Bayesian method for estimating the probability two classes from separate knowledge bases are equivalent given a sample of their instances. This measure gives us a consistent method of describing how certain we are a given correspondence exists. It can be used in cases where only small amounts of matched data are available, and in cases where very large amounts of data are available, it can help with scalability by allowing us to only consider a sample of the data.
- **Bayes-ReCCE** – a generate and test based search strategy which uses the similarity estimate to find Class Restriction complex correspondences between ontologies with matched instances. This search strategy restricts the potential restriction classes to those with pre-defined patterns of correspondences, allowing us to more accurately predict if the correspondence is true.
- A proposal for presenting to mapping creators the identified complex correspondences along with the evidence that supports their correctness. Knowing how well a correspondence is supported by the evidence examined allows users to make an informed choice when deciding if these correspondences can be reused, and we advocate should be incorporated into the mapping metadata.
- In addition to the use case of matching, the application of the approach presented is not only confined to LOD interlinking. For example, it could potentially be used to support: refinement of ontologies [35]; to improve quality through contributing to type prediction [36]; ontology learning and refinement [37][38]; query rewriting [39].

To evaluate the correctness of the class similarity measure and the search strategy, we demonstrate the use of ReCCE to detect complex correspondences between the GeoNames [9], DBpedia [4] and LinkedMDB [10] knowledge bases. We show that even with only a small number of matched instances (~15 per class), the Bayesian set similarity estimate employed by ReCCE provides comparable performance to using a Jaccard similarity measure based on thousands of matched instances.

The remainder of this article is laid out as follows. First we briefly describe the three types of complex correspondence patterns addressed by our method: Class by Attribute Value (CAV), Class by Attribute Type (CAT) and Class by Attribute Existence (CAE). These patterns are addressed as they commonly occur [7][8][40] between classes in Linked Open Data (LOD) datasets. Then we discuss extensional correspondence detection and describe a Bayesian approach to estimating the true Jaccard index of the instance sets of classes from separate knowledge bases. We then describe the Bayes-ReCCE algorithm which uses this estimate to detect complex correspondences. We describe an evaluation of Bayes-ReCCE using DBpedia, GeoNames and LinkedMDB, showing that a small sample of matched instances will allow us to find a set of complex correspondences with a median F1 score above 0.75 when compared with a gold standard. We then propose a set of metadata for presenting detected correspondences to users in a way which allows the users to make a more informed decision on which

correspondences to use. Next we present Related Work. Finally we compare our research with related work of other researchers in this area.

2 Complex Correspondence Patterns

Correspondence Patterns are a specific type of Ontology Design Pattern[1]. They serve a similar function to that of Design Patterns [11] in software engineering. The use of patterns provides several benefits. The use of templates to describe commonly occurring complex relationships between ontologies means that the relationships are described in a uniform manner and therefore are easier to interpret. Correspondence patterns provide two layered templates [6] which contain an abstract layer which describes the problem being solved using natural language terms. The second layer is known as a **grounding,** and describes a parameterised template solution to this problem expressed in a formal language – typically EDOAL [12]. By specifying values for the parameters in a grounding it is possible to create an instantiated grounding which can be used to perform a task such as query rewriting or instance translation [13][14] and perform other automated data mediation tasks.

However, the primary benefit of these patterns is that since they describe commonly occurring forms of correspondences, we can use them to build a search strategy for complex correspondences between two knowledge bases. Our search strategy, Bayes-ReCCE, considers three distinct patterns when searching for complex correspondences – Class by Attribute Value (CAV), Class by Attribute Type (CAT) and Class by Attribute Existence (CAE). These patterns are all types of restriction class correspondence pattern and share a similar structure. All specify a target class and a source class and a restriction feature, where the target class corresponds to the source class restricted to those instances which have the restriction feature. The patterns differ in what form the restriction feature takes. We use the notation $C|_f$ to denote the restriction of the class C to those instances with feature f.

2.1.1 Class by Attribute Value Correspondences

Class by Attribute Value Correspondences (CAV) occur when a named class in one ontology is equivalent to the subclass of a named class in a second ontology of exactly those instances which have a specified property set to a specified value. The restriction feature in this pattern takes the form of a pair, (p, v), where p is a property and v is a value – which could be a literal or a general resource. For example, given two ontologies describing wine, it is possible that one ontology may have a specific class for *BordeauxWine*, but the other does not. If, however, the other ontology has a class for wines in general, *Vin*, and instances of this class have an attribute which describes the region the wine comes from, *terroir*, then it may be appropriate to say that the target class *BordeauxWine* corresponds with the set of instances of *Vin* with the attribute *terroir* set to the value *Bordelais*.

[1] http://ontologydesignpatterns.org/

2.1.2 Class by Attribute Type

Class by Attribute Type Correspondences (CAT) occur when a named class in one ontology is equivalent to the subclass of a named class in a second ontology of exactly those instances which have a specified property set to an object with a specified type. The restriction feature in this pattern takes the form of a pair, (p, t), where p is a property and t is a type. An example of this pattern could be if we said the class *EUCitizen* corresponds with a restriction on the class *Person* to the instances which have the property *citizenOf* set to a country which has type *EUMemberState.*

The Inverse Class by Attribute Type (CAT^{-1}) correspondence pattern can also occur. This pattern is also specified by a restriction feature of the form (p, t), however in this case, the source class is restricted to those instances, i, which appear as the object of the of a subject, predicate, object relationship $\langle s, p, i \rangle$ where s has type t. For example the class *CarPart* might correspond with the class *MechanicalPart* restricted to the instances which are the object of a *hasPart* relationship for an instance of the class *Car*.

2.1.3 Class by Attribute Existence

Class by Attribute Existence Correspondences (CAE) occur when a named class in one ontology is equivalent to the subclass of a named class in a second ontology of exactly those instances which have a specified property but where the value is unimportant. For example the class *Deceased* may correspond with the class *Person* restricted to instances which have the property *diedOn* set to some value.

Within Bayes-ReCCE, by limiting our search to specific patterns of restriction class correspondence, we are able to employ a "generate and test" approach by creating candidate correspondences fitting the patterns, and then test how well these candidates are supported by the data in the knowledge bases. The method for generating candidates is discussed in detail in section 4.Before we discuss generating candidates we must define what we mean by a "valid" complex correspondence.

2.1.4 Prevalence of Complex Correspondences in Publicly Available Ontologies

Despite the long recognition of the utility of complex correspondences there are relatively few sets of published complex correspondences for public linked data, e.g. [7] [8] [40], and this makes estimation of the frequency and applicability of each correspondence type hard. However all the evidence we have collected to date points at the importance and relevance of CAV as the most frequently occurring pattern in both the data and the published mappings. CAT is the next most frequent pattern we observed and our method also includes support for CAE since this requires no additional implementation beyond CAT. We summarise this evidence in this section.

Looking at the work of Parundekar et al. [7] [8] we see automated discovery of 351 CAV correspondences between DBpedia and Geonames and an additional 5 correspondences with a

restriction class on either side of the relation. Geonames publish[2] a set of 32 correspondences between Geonames and DBpedia and all of these fit the CAV pattern. In section 5.1.2 we describe an additional 11 CAV mappings between Geonames and DBpedia that we discovered as part of our work. Additionally as described in section 5.1.1 we carried out a manual inspection of the relations between LinkedMDB and DBpedia and discovered 3 inverse CAT correspondences.

In order to get a better understanding of the distribution of complex correspondences we analysed the class restriction correspondences between YAGO (9 Jan 2012 core version) and DBpedia (version 3.7). Both ontologies are large and cover an almost identical domain (the contents of Wikipedia) however both ontologies are stuctured very differently with DBpedia having approximately 170 classes and YAGO 15,000. Hence there is strong reason to believe that it will be a rich source of complex correspondences – but this was chosen toinvestigate their distribution.

We randomly selected 50 classes from YAGO that had at least one instance and manually inspected them to determine the most appropriate correspondence type with DBpedia classes - equilivence, complex or none. Our results were 6 classes with no DBpedia equivilent, 12 direct equivilents, and 32 complex correspondences. 11 of the 32 complex correspondences were CAT and CAE was the next most common type. These results should be treated as necessarily subjective as the selection of the most appropriate complex correspondence type can have many possible criteria and the sample set and scenario is limited. Nonetheless it does provide an additional indicator for the importance of detecting CAV mappings in real-world datasets.

As part of this investigative work complex correspondences that go beyond the patterns of Scharffe [6] were detected, including the possibility of unions of patterns or the requirement to modify the target ontology (e.g. by adding a new relationship) but given the relative immaturity of complex correspondence detection research and the demonstrated applicability of CAV to known problems such as Geonmanes-DBpedia mappings we defer those issues to future work as they will not influence the basic metholdology of Bayes-ReCCE as presented in this paper.

3 Testing restriction class correspondence support

We consider an equivalence correspondence to be valid if the classes described in the correspondence have the same class extension sets. If we knew the exact set associated with the classes, the Jaccard index would provide us a method of measuring their similarity. If A and B are sets then the Jaccard index of A and B is:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

With $0 \leq J(A,B) \leq 1$, and $J(A,B) = 1$ iff the sets of A and B are identical.

[2] Available at (25/2/2016) http://www.geonames.org/ontology/mappings_v3.01.rdf

However, the extension sets of a given class are not generally known. Often we only know of a few instances in the set. To understand how the Jaccard similarity measure relates to classes in an ontology, we must look at the underlying description logic these classes are described in. Typically the Linked Data KBs have associated OWL TBoxes which describe them. OWL uses an extended form of the $\mathcal{ALC}$ Description Logic. In $\mathcal{ALC}$, N_C is the set of atomic concepts, and N_R is the set of roles with $N_R \cap N_C = \emptyset$. Each atomic concept is a concept, and additional (complex) concepts can be constructed by combining concepts and roles. Each $\mathcal{ALC}$ concept C can be associated with a set by means of an *interpretation*, $\mathcal{I}$, consisting of an **interpretation** domain $\Delta^{\mathcal{I}}$, and an interpretation function, $\cdot^{\mathcal{I}}$, which maps C to a set $C^{\mathcal{I}} \subset \Delta^{\mathcal{I}}$. In general we, do not know the exact set associated with a given class. The set of all names is denoted N_I, and for $a \in N_I$ and $C \in N_C$ the notation $C(a)$ is used to assert that a is an instance of C.

To allow interpreting ABoxes, Hellmann et al. extend the definition of interpretation. This extended definition [15] states that for the class C and individual a, if $\mathcal{K} = \{\mathcal{T}, \mathcal{A}\}$ is an ontology with TBox $\mathcal{T}$ and ABox $\mathcal{A}$, then a is an instance of *C* with respect to $\mathcal{K}$, denoted $K \vDash C(a)$ iff for any model $\mathcal{I}$ of $\mathcal{K}$ we have $a^{\mathcal{I}} \in C^{\mathcal{I}}$. The retrieval of class *C* with respect to $\mathcal{K}$, is defined as: $R_{\mathcal{K}}(C) = \{a | a \in N_I, \mathcal{K} \vDash C(a)\}$, with N_I being the set of all names. More simply, retrieval is the set of all named instances that the knowledgebase $\mathcal{K}$ states are members of class C.

Therefore we can observe that some instances are members of the set, but due to the Open World Assumption, we cannot say that these instances comprise the whole set, unless we have been explicitly told this is the case. In turn this means that if we retrieve the known members of two classes and use the Jaccard to calculate the similarity we cannot be certain that this measure is correct, as instances from the extension set could be missing due to their membership being *unknown.*

When comparing classes between ontologies which have been mapped to one another, we face further difficulties in knowing the extension sets. We cannot, in general, say that we know all mappings between two ontologies, and this introduces a further level of unknowns. For two given classes, not only do we not generally know all the members of each of the classes' extension sets, we do not usually know for certain all the mappings between the instances in the extension sets. To explain this difficulty, we extend the concept of retrieval to **mapped retrieval**.

Suppose $\mathcal{K}_s$ and $\mathcal{K}_t$ are ontologies, our source and target ontologies, C is a class described in $\mathcal{K}_t$, and M is a set of mappings of the things described in $\mathcal{K}_s$ and $\mathcal{K}_t$. Then the mapped retrieval

$$R_{\mathcal{K}_s|\mathcal{K}_t,M}(C) = \{a | (a, a') \in M, \mathcal{K}_s \vDash \top(a), \mathcal{K}_t \vDash C(a')\}.$$

That is the set of all things which $\mathcal{K}_s$ states are named things and which are mapped, via M, to a thing which $\mathcal{K}_t$ states is a thing of type C. Using mapped retrieval introduces another step where missing information can become an issue. In order for the individual a to be include in

$R_{\mathcal{K}_s|\mathcal{K}_t,M}(C)$, we require that the mapping (a, a') be included in M and that $\mathcal{K}_t \vDash C(a')$. If the mapping is unknown, or $\mathcal{K}_t$ does not tell us that a' is an instance of C, then a will not be included in the mapped retrieval. Note that the set of classes mapped to are the potential restriction classes rather than just those defined in the target ontology itself. Hence if the presence or value of an attribute (property) is an indicator for a suitable restriction class, i.e. a CAT or CAE correspondence, then it will be included in the mapped recall.

We cannot, in general, know the elements of the set associated with a given class. When comparing the elements of the sets associated with classes contained in separate ontologies we are limited to the mapped retreival, and this is liable to be a much smaller set than the true intersection of the classes' extension sets. We are forced to work with incomplete information, and this introduces an element of uncertainty. It is important to quantify this uncertainty in a manner which can be clearly explained. Bayesian statistics provides us with a well-established method for doing this [16] The following section describes how a Bayesian prior probability can be used to estimate the overlap in the sets associated with two classes.

3.1 Using a beta binomial distribution to estimate the similarity of two Classes

For classes C and D, we cannot generally calculate the true value of the size of the intersection or union of the sets associated with the classes, which in turn means that we cannot calculate $J(C, D)$ directly. Therefore, we must estimate it. The Jaccard index measures the proportion of instances in the union of two sets which are also in the intersection of the sets. Suppose we name this proportion θ, then for n instances $x_{1..n} \in C \sqcup D$, the probability that k of these instances are also members of $C \sqcap D$ can be calculated using the binomial distribution

$$p(k|\theta, n) = \binom{n}{k} \theta^k (1-\theta)^{n-k}$$

Using Bayes Theorem, we can say that

$$p(\theta|k, n) = \frac{p(k|\theta, n)P(\theta)}{P(k)}$$

$$\propto p(k|\theta, n)P(\theta)$$

$$= L(\theta|k, n)$$

Where $L(\theta|k, n)$ is known as the likelihood function. Using (1), one option for estimating θ would be to use the maximum likelihood estimator (MLE). This is the value of θ which maximises $L(\theta|k, n)$. Assuming all values of θ are equally probable, then the MLE of $L(\theta|k, n)$ is $\hat{\theta} = \frac{k}{n}$. Using this MLE we can then decide that C and D are equivalent if $\hat{\theta} > 1 - \varepsilon$.

While easy to implement, this approach does not allow us to say anything about how strongly we believe the results it produces. If we look at a single instance and see that it is a member of both $C \sqcap D$ and $C \sqcup D$, then $n = 1$, $k = 1$, and the MLE is also 1. This MLE is slightly higher than the one obtained from seeing a 1,000 instances of $C \sqcup D$, 999 of which are members $C \sqcap D$. Clearly, the evidence based on a thousand instances should hold much more weight than evidence based on a single instance, but MLE does not give us a clear way of accounting for this difference in weight of evidence.

Instead, the method we employ in this paper is a Beta Binomial conjugate prior distribution. The following paragraphs give a brief description of how this distribution is derived, and how it can be used. For a thorough explanation of this distribution see [17].

Assume that θ is not fixed but that it is a variable drawn from a Beta distribution:

$$\theta \sim Beta(\alpha, \beta)$$

$$p(\theta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)}$$

Where $B(\alpha, \beta)$ is Euler's beta function. Then, as $p(\theta|k,n) \propto p(k|\theta,n)p(\theta)$ we have:

$$p(\theta|k,n) \propto \binom{n}{k} \theta^k (1-\theta)^{n-k} \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

$$\propto \theta^{k+\alpha-1}(1-\theta)^{n-k+\beta-1} \quad (1)$$

Combining equation (2) **Error! Reference source not found.** with the fact that, as it is a probability density function, $\int_0^1 p(\theta|k,n)d\theta = 1$, shows us that if we observe n instances of $C \sqcup D$, k of which are instances of $C \sqcap D$, then

$$\theta \sim Beta(k+\alpha, n-k+\beta)$$

The values α and β are known as the prior parameters and allow us to encode or prior belief on what the value of θ should be. Choosing appropriate prior parameters is generally done by intuition, by past experience, or by fitting to experimental data from similar situations (Shultis & Eckhoff, 1979). For example, suppose we wish to test a CAV correspondence between classes A and B with the restriction feature (p, v). If we do not want to make any strong assumptions, setting $\alpha = \beta = 1$ gives the **uninformed prior,** which assumes that all values of $\theta \in [0,1]$ are equally probable. If however, from prior experience we feel that restriction features based on the

property p are not common, we might set $\alpha = 1, \beta = 4$ which gives us a prior distribution where low values of θ are more probable. This prior distribution is illustrated in Figure 1 (a). If we then observe 10 instances of class A, 9 of which belong to the restriction class formed by class B and restriction property (p, v), we can infer the posterior distribution shown in Figure 1 (b). Here the distribution generated using the uniform prior shows higher values of θ to be more probable than the distribution generated using $\alpha = 1, \beta = 4$ as the prior parameters. If we observe 100 instances, 95 of which are in the intersection of the classes, then we can infer the distribution shown in Figure 1 (c).

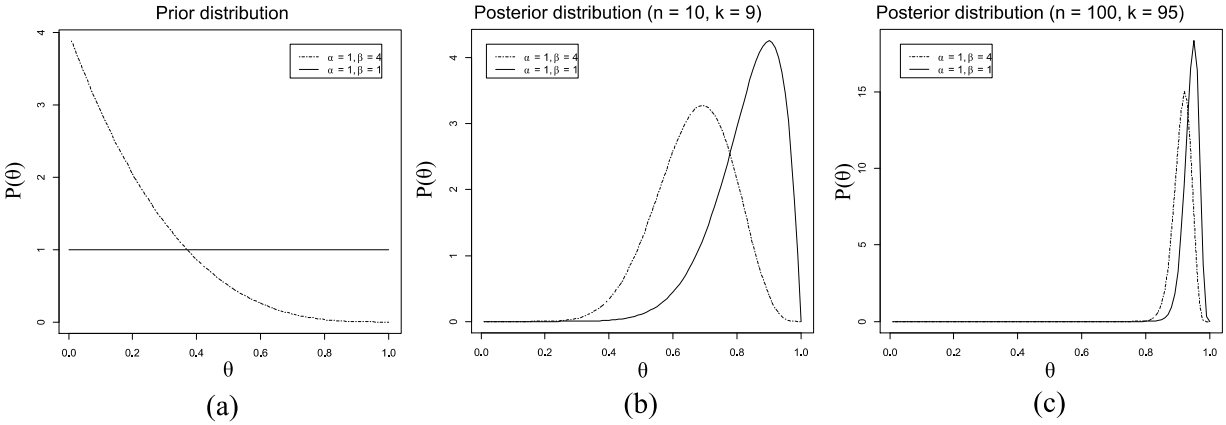


Figure 1. Prior and posterior distributions with beta parameters $\alpha = 1, \beta = 1$, and $\alpha = 1, \beta = 4$.

Treating θ as a random variable with probability density function instead of just a likelihood, allows us to make a much stronger statement about our belief. Instead of simply finding the MLE, and testing if it is higher than the $1 - \varepsilon$ cutoff, we can calculate $P(\theta > 1 - \varepsilon | k, n, \alpha, \beta)$. Larger values of k and n will provide a distribution with less variance. Using a conjugate prior means we can understand how the size of our sample of matched instances affects the strength of our gives us a formal method for balancing our prior beliefs against the extensional evidence we have examined. It is important to note that incompleteness of instance links is not the only possible source of bias in an extensional mapping detection technique such as this – for example there could be cultural, technical or social reasons for the distribution of existent links in the datasets. However it is important to note that there are two mitigating factors in our suggested deployment approach for Bayes-ReCCE (1) since only very small numbers of links are required it is anticipated that it will be a human-supervised process that selects or creates the input links and (2) Bayes-ReCCE only makes ranked recommendations on possible complex correspondences and these will need to be post-processed, in a semi- or fully-supervised tool-chain for the best possible mapping results. Having defined the patterns of correspondence we are searching for, and the criteria which we use to determine if a correspondence is valid, we now describe our search algorithm in detail.

4 Complex correspondence learning algorithm

In this section we describe the steps in the Bayesian Restriction Class Correspondence Estimation (Bayes-ReCCE) algorithm which detects complex correspondences fitting the CAV, CAT, and CAE patterns described in Section 2. Bayes-ReCCE uses a generate and test approach to detecting complex correspondences. There are three major steps to this approach, which are illustrated in Figure 2. Bayes-ReCCE requires a set of elementary class correspondences between the ontologies as input. There are many tools and approaches available [18,19,20,21,22] to assist in this task of finding elementary correspondences, so we assume that creating this input set is a tractable problem. For each elementary correspondence Bayes-ReCCE will create a set of candidate complex correspondences by selecting target classes from the target ontology and restriction features from the source ontology. In the second step, each of the candidate correspondences is tested to estimate the probability that the target and restriction classes in the correspondence are equivalent, rejecting any that do not meet some configurable minimum probability. Finally, in a refinement step the remaining correspondences are examined to remove any that conflict with other correspondences. Each of these steps are described in detail below.

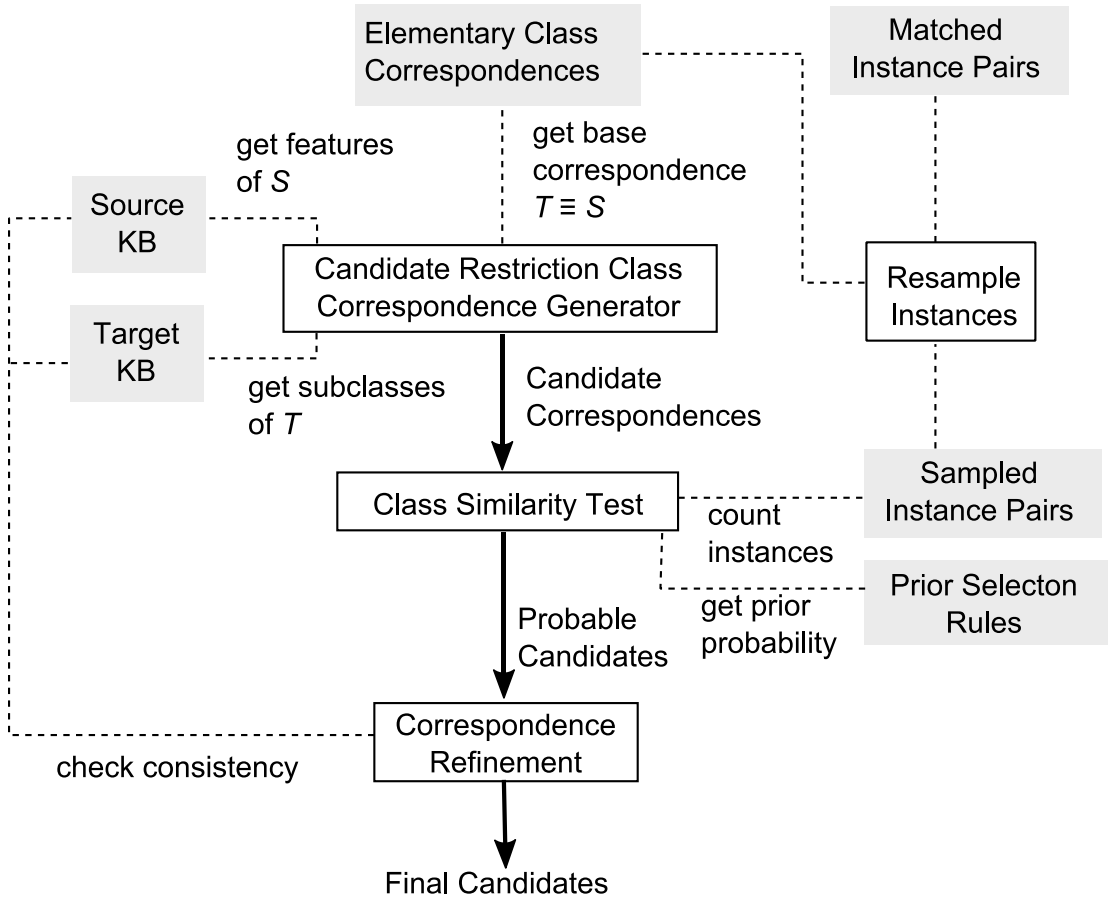


Figure 2. Generate and Test approach to detecting complex correspondences

4.1 Candidate Restriction Class Correspondence Generation

To detect these complex correspondences between a target knowledge base, $\mathcal{K}_t$, and a source, $\mathcal{K}_s$, our Bayes-ReCCE algorithm requires a set of elementary class correspondences between $\mathcal{K}_t$ and $\mathcal{K}_s$, and a set, M, of pairs of matched instances from $\mathcal{K}_t$ and $\mathcal{K}_s$. For each input correspondence of the form $T \equiv S$, Bayes-ReCCE will search for correspondences of the form $T_i \equiv S|_f$, where T_i is a subclass of T and f is a restriction feature. Let $T_{img} = R_{\mathcal{K}_t|\mathcal{K}_s,M}(T_i)$, be the set of instances of S which match an instance of T_i. Then the features can be generated using the following SPARQL pseudo-queries:

To detect CAV features

```
SELECT DISTINCT ?p ?v
WHERE {
   ?i ?p ?v.
   ?i ∈ T_img .
}
```

To detect CAT features

```
SELECT DISTINCT ?p ?t
WHERE {
   ?i ?p ?o.
   ?o rdf:type ?t
   ?i ∈ T_img .
}
```

To detect CAT^{-1} features

```
SELECT DISTINCT ?p ?t
WHERE {
   ?s ?p ?i.
   ?s rdf:type ?t
   ?i ∈ T_img .
}
```

To detect CAE features

```
SELECT DISTINCT ?p
WHERE {
   ?i ?p _.
   ?i ∈ T_img .
}
```

4.2 Class Similarity Test

During this step, candidate correspondences of the form $T_i \equiv S|_f$ are tested to estimate the probability that T_i really is equivalent to $S|_f$. This estimation is performed using the Bayesian method described in Section 3.1. We are estimating the probability that $\theta = J(T_i, S|_f)$ is greater than $\varepsilon \in [0,1]$. To do this we compare the sets T_{img} with the set S_M which consists of the instances in the mapped retrieval of S, that is the instances of S which are matched to an instance of T. It is necessary to use S_M because the retreival of S may contain instances of T_i which are not included in T_{img} due to not being included in the mapping set. See Figure 3. Using S_M we calculate $S_f = S_M|_f$, the set of matched instances of S which have feature f. We can then use the sizes of the intersection and union of T_{img} and S_f combined with our priors α and β to estimate $P(\theta > \varepsilon)$. If this probability is lower than a specified cut off, p_{MIN}, we reject the correspondence candidate.

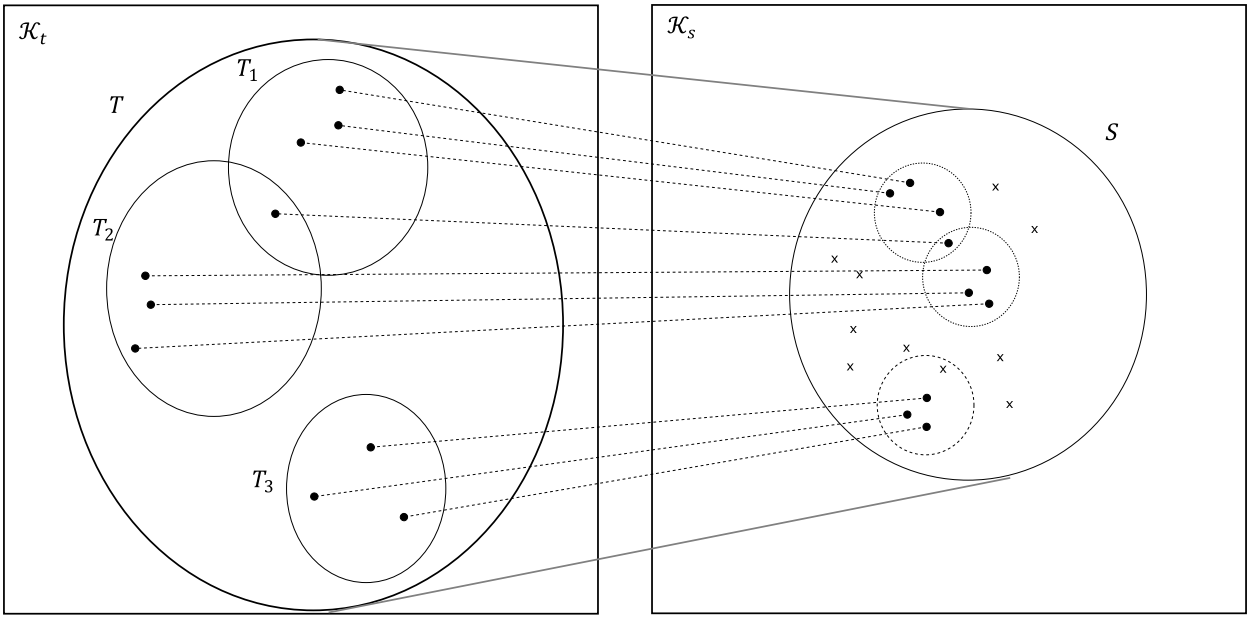


Figure 3. Mapping each of the T_i sets to subsets of S.

4.3 Correspondence Refinement Step

Initial implementations of our approach were observed to over generalise the forms of correspondence detected. For example, detecting that *film directors* correspond with people who had directed any creative work, not just films. To remedy this, we use a set of heuristic rules to find the most specific correspondence for each target class. We assume that CAV correspondences are the most specific, CAT are less specific, and CAE the least. We also assume that if class A is a subclass of B, then a CAT correspondence conditioned on A is more specific than a CAT conditioned on B. The set of rules we use to remove the over-general mappings is as follows:

1. If $T_i \equiv S|_{f_1}$ is a CAE correspondence and there exists a CAV or CAT correspondence $T_i \equiv S|_{f_2}$ which uses the same restriction property then $T_i \equiv S|_{f_1}$ is removed from the set of candidate correspondences, as $S|_{f_2}$ is a more specific class than $S|_{f_1}$.
2. If $T_i \equiv S|_{f_1}$ is a CAT correspondence and there exists a CAT correspondence $T_i \equiv S|_{f_2}$ where the restriction type specified in f_2 is more specific than the restriction type specified in f_1, then $T_i \equiv S|_{f_1}$ is removed, as $S|_{f_2}$ is a more specific class than $S|_{f_1}$.
3. If $T_i \equiv S|_{f_1}$ is a CAT correspondence and there exists a CAV correspondence $T_i \equiv S|_{f_2}$ where the restriction value specified in f_2 is an instance of the restriction type specified in f_1, then $T_i \equiv S|_{f_1}$ is removed, as again, $S|_{f_2}$ is a more specific class than $S|_{f_1}$.

Once these rules have been applied, we are left with our final set of candidate correspondences.

4.4 Resampling the set of instance matches

The number of features examined while searching for the correspondences is a function of the number of instances contained in knowledge bases. As the size of the ABox increases, the number of features that need to be examined can be very large. If a representative sample of the instances is used instead, it should allow us to calculate our class similarity measure to a reasonable degree of accuracy without needing to examine as many features.

Therefore we include an additional pre-processing step in our search algorithm where we resample the set of matched instances. For each target class T_i that we will be evaluating we select at most N match pairs from M which refer to instances of T_i and add the pairs to the set M'. We then use M' in place of M when evaluating class similarity.

The complete Bayes-ReCCE algorithm is listed in Algorithm Listing 1. The following section describes an evaluation which was carried out to demonstrate that Bayes-ReCCE is capable of detecting complex correspondence in real-world Linked Data knowledge bases, and to investigate the effects of matched instance sample size on its performance.

Algorithm Listing 1: Bayes-ReCCE

Input:
C, a set of elementary class correspondences
$\boldsymbol{M}$ a set of matched instances,
ε, p_{MIN}, N
prior_rules, a set rules for selecting Bayesian parameters.

Output: $\boldsymbol{RC}$, a set of restriction correspondences

$\boldsymbol{RC} \coloneqq \{\}$
$\boldsymbol{M}$ = `resample(`$\boldsymbol{M}, N$`)`
foreach $(T, S) \in \boldsymbol{C}$ **do**
  $S_M \coloneqq instances\ in\ R_{\mathcal{K}_s}(S)\ matched\ to\ T$
  foreach $T_i \sqsubset T$ **do**
    $T_{img} \coloneqq R_{\mathcal{K}_t|\mathcal{K}_s,\boldsymbol{M}}(T_i)$
    foreach $f \in$`features(`T_{img}`)` **do**
      $(\alpha, \beta) \coloneqq$ `selectPrior(`f**,** prior_rules`)`
      $S_f \coloneqq S_M|_f$
      $k \coloneqq |T_{img} \cap S_f|$
      $n \coloneqq |T_{img} \cup S_f|$
      if $P(\theta > \varepsilon | k, n, \alpha, \beta) > p_{MIN}$ **then**
        add $T_i \equiv S|_f$ to $\boldsymbol{RC}$
  `removeConflicts(`$\boldsymbol{RC}$`)`
return $\boldsymbol{RC}$

5 Empirical Evaluation

The following section describes an evaluation of the performance of Bayes-ReCCE on large, real world Linked Data knowledge bases, and assesses the effect of matched instance sample size on its ability to accurately detect complex correspondences. For the evaluation we manually created two sets of gold standard complex correspondences, one set with DBpedia as the source ontology and LinkedMDB as the target ontology, and a second set with GeoNames as the source and DBpedia as the target. We then use Bayes-ReCCE to automatically detect complex correspondences between the ontologies, using a range of values of N in the resample step described in Algorithm Listing 1. The correspondences produced each time are compared with our gold standard correspondences to produce an F1 score, and this process was carried out 20 times per value of N to estimate the average F1 score as a function of N, the matched instance sample size. Our hypothesis is that:

For some value of N a random sample of N instances per target class will allow us to find a set of complex correspondences with a median F1 score above 0.75 when compared with the gold standard.

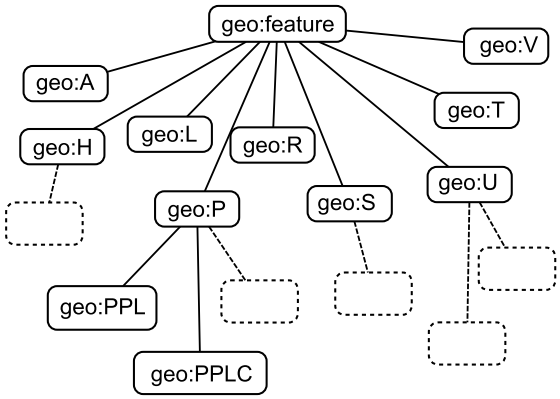
If true this would indicate that for this value of N, a sample of instances would have a one in two chance of the mean of the precision and recall scores being above 0.75.

In the remainder of this section we first describe the knowledge bases we use in the evaluation – DBpedia, GeoNames and LinkedMDB – and a gold standard set of complex correspondences which exist between them. We then show the results of using Bayes-ReECCE to find correspondences between the ontologies using samples of matched instance data.

5.1 Datasets

DBpedia is a large, cross-domain knowledge base containing information extracted from Wikipedia. The 2014 releases used in this evaluation contains about 4.5 million things, approximately 4.2m of which are classified in a consistent ontology. This includes details of approximately 1,445,000 persons and 735,000 places. The DBpedia ontology contains 685 classes which form a subsumption hierarchy and 2,795 different properties. DBpedia is one of the central resources in the linked open data cloud, and as such it is highly connected with many other knowledge bases.

GeoNames is a geographical knowledge base which describes over 9 million unique geographical features. The GeoNames class hierarchy is extremely flat, and all features in GeoNames simply have the class *geo:Feature*. Categorisation in GeoNames is achieved using two properties *geo:featureClass* which can take one of 9 values, and geo:featureCode which can take one of 645 values. The combination of *geo:featureClass* and geo:featureCode can be seen as forming a two level hierarchy which is illustrated in Figure 4.



Geoname Feature Classes	
Class	Meaning
A	country, state, region,...
H	stream, lake, ...
L	parks,area, ...
P	city, village,...
R	road, railroad
S	spot, building, farm
T	mountain,hill,rock,...
U	undersea
V	forest,heath,...

Geoname Feature Codes	
Code	Meaning
PPL	populated place
PPLC	capital of a political entity

Figure 4. A partial illustration of the GeoNames classification hierarchy

One important distinction between this hierarchy and one provided using classes is that while each individual in general can belong to multiple classes, individuals in GeoNames can only have a maximum of one value each for their geo:featureClass and geo:featureCode properties. In addition GeoNames feature codes and classes use cryptic combinations of letters and it is not immediately obvious what they represent. For example the code "PPLC" refers to the capital of a political entity.

LinkedMDB describes approximately 66,500 instances from the domain of cinema. It has a flat hierarchy of 53 classes.

5.1.1 Relationship between DBpedia and LinkedMDB

There are 30,354 instance links between DBpedia and LinkedMDB, however there are no published correspondences between the classes used in LinkedMDB and those in the DBpedia ontology. The class *lmdb:film* obviously corresponds with the class *dbpedia-owl:Film*, and in initial inspection it might appear that *lmdb:actor* corresponds with *dbpedia-owl:Actor* and *lmdb:director* corresponds to *dbpedia-owl:MovieDirector*. Further inspection reveals that the correspondences for actors and directors are not so clear cut. There are no individuals in the 2014 release of DBpedia which have type *dbpedia-owl:MovieDirector*, and of the 3838 instances of *lmdb:actor* that are matched to instances in DBpedia, only 49 are identified as having type *dbpdia-owl:Actor*. In addition to this extensional difference, actors in LinkedMDB are specifically film actors, while actors in DBpedia can also include television, theatre and radio actors, so despite having very similar names, the classes are intentionally different.

Instead of using elementary correspondences we can use complex correspondences to provide a more accurate alignment between DBpedia and LinkedMDB. Through manual analysis, we discovered three CAT^{-1} correspondences with DBpedia as the source ontology LinkedMDB as the target ontology. These correspondences are

- $lmdb{:}actor \equiv dbpedia\text{-}owl{:}Person|_{dpbedia\text{-}owl:starring \text{ of a } dbpedia:Film}$
- $lmdb{:}director \equiv dbpedia\text{-}owl{:}Person|_{\text{dpbedia-owl:director of a dbpedia:Film}}$
- $lmdb{:}producer \equiv dbpedia\text{-}owl{:}Person|_{dpbedia\text{-}owl:producer \text{ of a } dbpedia:Film}$

That is LinkedMDB actors correspond to instances of Person in DBpedia who are listed as a star of a film instance, directors in LinkedMDB are listed as the director value of a film in DBpedia, and similarly producers in LinkedMDB are listed as the producer value of a film in DBpedia. The average Jaccard index for the classes in these correspondences is 0.78.

5.1.2 Relationships between DBpedia and GeoNames

There are approximately 425,000 instance matches between DBpedia and GeoNames[3]. In addition the GeoNames website provides a set of class correspondences from GeoNames to a

[3] Available from http://wiki.dbpedia.org/Downloads39#links-to-geonames, accessed 28/11/2014

number of other knowledge bases including 32 correspondences with DBpedia classes. Each of these 32 correspondences follows the CAV pattern with GeoNames as the source and DBpedia the target. In each correspondence the target class is a subclass of *dbpedia-owl:Place*, the source class is *geo:Feature*, and the property geo:featureCode is used in the restriction feature. Through manual inspection of the ontologies we discovered an additional 11 CAV correspondences (see Table 1).

Table 1: Class by Attribute Value correspondences between DBpedia and GeoNames

Named Class	**Restriction Class**
dbpedia:Park	rdf:type (X, geo:feature) ∧ geo:featureCode(X, "L.PRK")
dbpedia:Mountain	rdf:type (X, geo:feature) ∧ geo:featureCode(X, "T")
dbpedia:BodyOfWater	rdf:type (X, geo:feature) ∧ geo:featureClass(X, "H")
dbpedia:Country	rdf:type (X, geo:feature) ∧ geo:featureCode(X, "A.PCLI")
dbpedia:Island	rdf:type (X, geo:feature) ∧ geo:featureCode(X, "T.ISL")
dbpedia:Airport	rdf:type (X, geo:feature) ∧ geo:featureCode(X, "S.AIRP")
dbpedia:Hospital	rdf:type (X, geo:feature) ∧ geo:featureCode(X, "S.HSP")
dbpedia:Bridge	rdf:type (X, geo:feature) ∧ geo:featureCode(X, "S.BDG")
dbpedia:Lake	rdf:type (X, geo:feature) ∧ geo:featureCode(X, "H.LK")
dbpedia:River	rdf:type (X, geo:feature) ∧ geo:featureCode(X, "H.STM")

The majority of which use *geo:featureCode* in their restriction feature, but in addition we include the following correspondences which use geo:featureClass:

- $dbpedia\text{-}owl:BodyOfWater \equiv geo:Feature|_{geo:featureClass = H}$
- $dbpedia\text{-}owl:ArchitecturalStructure \equiv geo:Feature|_{geo:featureClass = S}$

As well as the following correspondence using parent feature:

- $\text{dbpedia-owl:Continent} \equiv \text{geo:Feature}|_{\text{geo:parentFeature = http://sws.geonames.org/6295630/}}$

This correspondence may appear incorrect at first glance, however http://sws.geonames.org/6295630/ refers to the Earth, all continents have this feature as their parent, and they are they are the only features to do so.

Inspecting the extensional similarity of the classes in the correspondences listed on the GeoNames website revealed that some are quite dissimilar. The Jaccard index for each of the target classes and their corresponding restriction classes ranged from 0, for atolls, to 0.9, for hospitals and the mean value was 0.43.

To create a gold standard set of complex correspondences we combined the 32 published correspondences with the 11 we discovered manually[4].

5.2 Results

In our evaluation we use a p_{min} value of 0.5 and an ε value of 0.5 – that is selecting correspondences where there is at least a 50% chance that the classes in the correspondence overlap by 50% or more. This choice of parameters promotes recall over precision. The prior selection rules used in this evaluation were very simple. Any correspondence using owl:sameAs received prior parameters $\alpha = 1, \beta = 9$, and all others used the uninformed prior, $\alpha = \beta = 1$. Applying Bayes-ReCCE to DBpedia and LinkedMDB, the median F1 score very quickly converges to 1 as the number of matched instances is increased. Figure 5 shows the mean F1 score for correspondences found between DBpedia and LinkedMDB as a function of matched instance sample size. The scores are quite variable at first, but this is partially due to the small number of cases in the gold standard. With a sample of 10 matched instances per target class, the interquartile range is above 0.75, and by 30 instances the F1 score has converged to 1.

[4] Available form http://scss.tcd.ie/~bwalshe/correspondences/geonames_dbpedia_correspondences.rdf

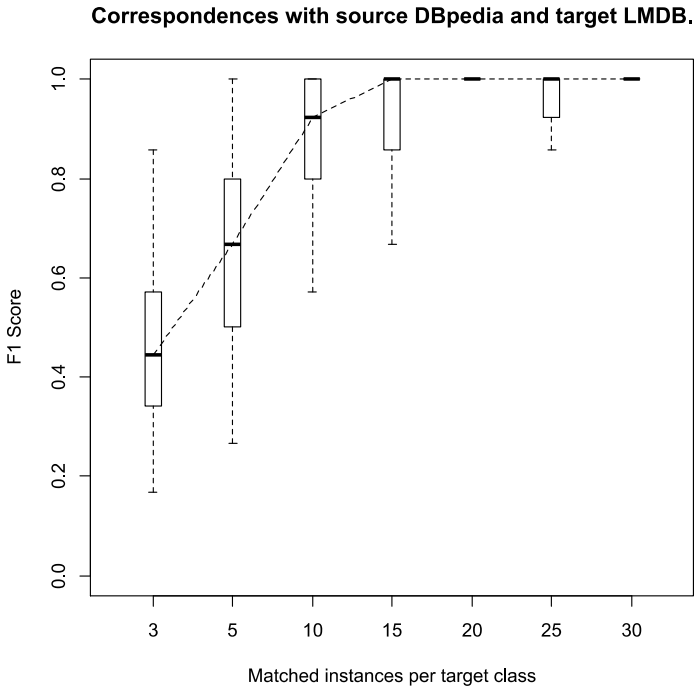


Figure 5. Mean F1 scores for correspondences found between DBpedia and LinkedMDB. Boxes show interquartile range, whiskers show the 95% confidence interval, and circles represent outliers.

Figure 6 shows the resulting F1 scores from applying Bayes-ReCCE to GeoNames and DBpedia. Here the F1 score is not as high as with the IMBD to DBpedia mappings. We would like to highlight however, that large amounts of matched instances were not required to achieve reasonable performance. With only 5 instance matches per target class, the 95% confidence interval for the mean F1 score was between 0.58 and 0.62. This compares well with the mean score when using 50 instances, which lies somewhere between 0.63 and 0.65 with 95% confidence. Figure 7 compares the performance of using a Bayesian estimator of similarity instead the directly observed Jaccard score. Here the same training set was used for each set of training data, and the ReCCE algorithm applied twice – once using the Bayesian Estimator described in section 3.1 and once using a direct Jaccard to score class similarity. Here we can see that the using a Bayesian estimator consistently scores a higher F1 score ($F1_{Bayes}$) than using a direct Jaccard ($F1_J$)

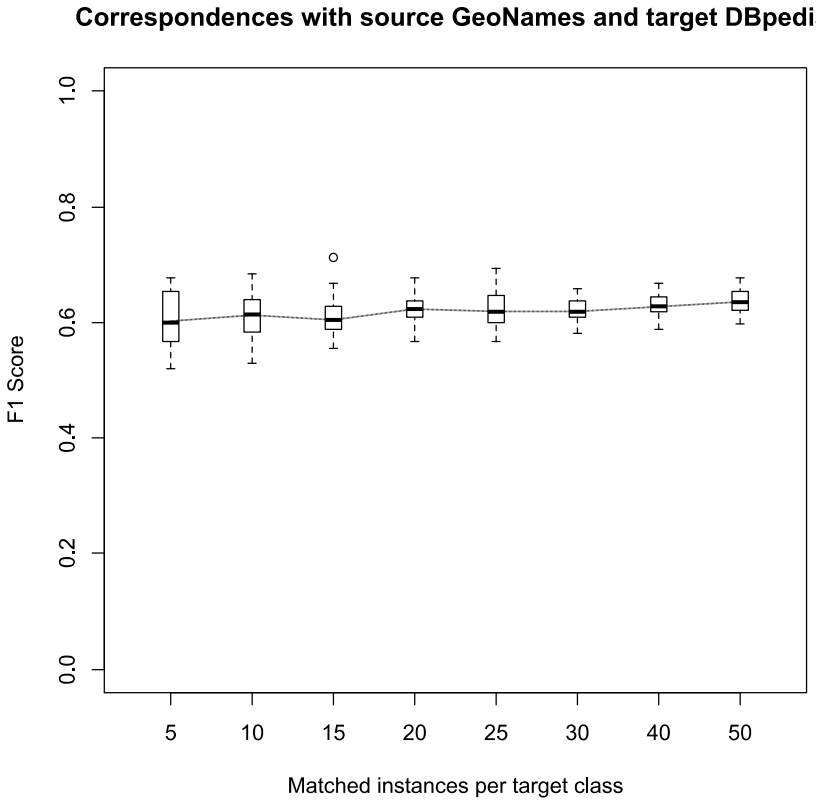


Figure 6. Mean F1 scores for correspondences found between GeoNames and DBpedia. Boxes show interquartile range, whiskers show the 95% confidence interval, and circles represent outliers.

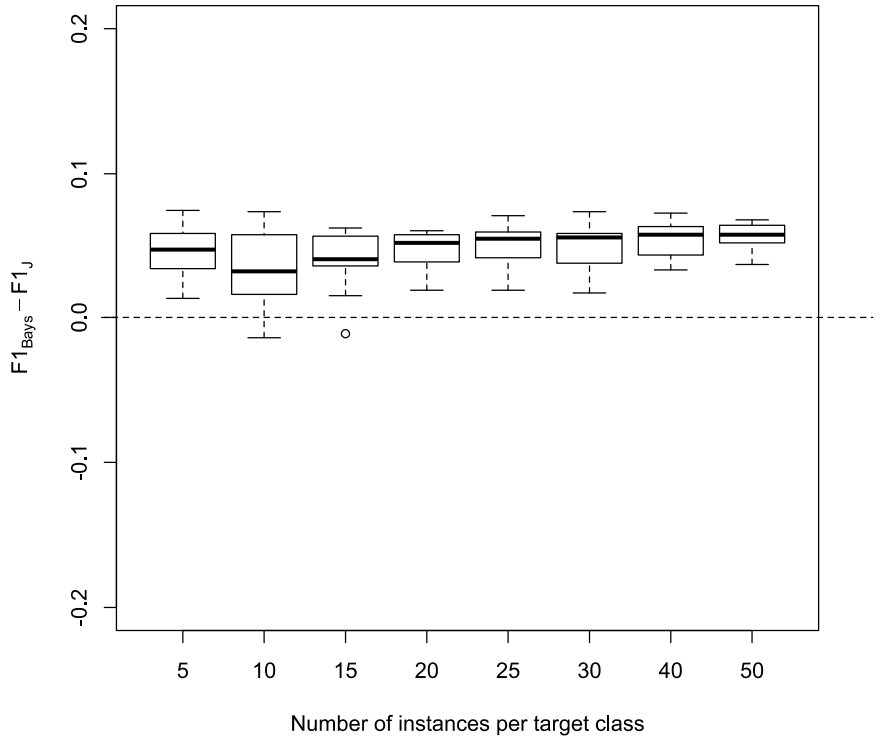


Figure 7 Precision and Recall with only 3 instance matches per target class.

5.3 Analysis

The Bayes-ReCCE algorithm demonstrated good performance with small samples of matched instances. With only 15 instances per target class the median F1 scores were 1.0 and 0.7 for correspondences between DBpedia and LinkedMDB, and GeoNames and DBpedia respectively. At 30 instances Bayes-ReCCE performed perfectly, detecting correspondences between DBpedia and LinkedMDB, and at 50 instances, the median F1 score for the correspondences detected between GeoNames and DBedia was above 0.63, with a 95% chance that the mean lies between 0.63 and 0.65. Performance for correspondence detection between DBpedia and LinkedMDB was better, and this is most likely as the gold standard correspondences for these ontologies had higher Jaccard indexes for their corresponding classes.

The benefits of using an extensional approach to class similarity are clear. Bayes-ReCCE did not return dbpedia-owl:Actor ≡ lmdb:actor, or dbpedia-owl:MovieDirector as final candidate correspondences[5]. As discussed in section 5.1.2, DBpedia actors are a broader class than LinkedMDB actors, and there are no instances of dbpedia-owl:MovieDirector in DBpedia currently. Bayes-ReCCE was also able to find correspondences between DBpedia and GeoNames despite the cryptic codes used in GeoNames.

[5] Bayes-ReCCE does consider these as initial candidate correspondences as they, technically, are a form of CAV which use rdf:type as the restriction property.

6 Correspondence Explanation and Reuse

The issue of using the correct predicate to describe the relationship for classes that have been matched is a difficult one. This can be seen in the "sameAs problem" [23] where *owl:sameAs* is widely used by Linked Data when it is not appropriate. Many of the problems, that apply to incorrect use of owl:sameAs also apply to owl:equivalentClass – such as the relationship being transitive, which has the potential to create "the semantic equivalent of mushy peas", if it is over used [23]. Other predicates are available such as *skos:exactMatch* and *skos:closeMatch*. The SKOS reference states that "a *skos:closeMatch* link indicates that two concepts are sufficiently similar that they can be used interchangeably in **some** information retrieval applications". However, if we were to publish a set of correspondences which we felt represented a close match, and we were to use this predicate, another user on inspecting the link might ask – "are these classes sufficiently similar for **my** information retrieval task?". The link on its own does not provide enough information to make an informed decision.

This problem of explaining matches has been highlighted as one of the ten most pressing challenges in semantic matching [13], [24]. The challenge is to provide explanations in a simple yet clear and precise way to facilitate informed decision making. Currently correspondences are described with a specific predicate and an optional confidence value, which can vary between 0 and 1, but which has no commonly agreed meaning.

We propose that instead of describing the match with a specific predicate, we include a summary of the evidence used to select the match and allow the users of the correspondence to decide for themselves if this evidence is sufficient for their particular information retrieval task. This evidence could also be included in the metadata of a mapping, that we have motivated and described in separate research [21].With each correspondence, $A \equiv B$, we include the following information

- k, the number of individuals we have observed that are instances of both A, and B.
- n, the number of individuals we have observed that are instances of either A, or B.
- α and β, the parameters of the prior we used when selecting this correspondence.
- j, the minimum Jaccard index we require for a match.
- p, the probability that the true Jaccard index is greater than j.

With this information, a consumer of the correspondence can see how we chose the correspondence, and can calculate for themselves the probability of the correspondence having a different minimum Jaccard index. This would enable reliable, informed decision making about correspondence re-use. For example, if we publish a set of correspondences which we believe to have a minimum Jaccard index of 0.5 or more, but a user requires correspondences which they believe to have a minimum Jaccard of 0.9, they can use the values provided to calculate the probability of the Jaccard being over 0.9 for each of our correspondences, and reject any that do not have a sufficiently high probability.

7 Comparison with other approaches

As techniques for automating the process of discovering complex correspondences is still an emergent topic, there are relatively few published approaches addressing the problem. The two most directly comparable approaches are an extensional approach described by Parundekar et al. [7] and a pattern matching approach described by Ritze et al. [5].

Ritze et al.'s approach is similar to Bayes-ReCCE in that it uses correspondence patterns to detect complex correspondences. Their approach differs in that it does not use any extensional information to perform pattern matching. Instead it analyses the structure of the ontologies and uses a series of string manipulation operations to test if the words used in the names classes and properties in one ontology can be re-arranged to form names similar to the ones used in the second ontology. The advantage of this approach is that it does not require any matches instances in order to be able to detect the complex correspondences. The disadvantage is that, in its current form, it is only able to detect CAV correspondences with restriction features based on Boolean values. Furthermore, if the approach was extended to be capable of detecting general CAV correspondences, it would still require the values to be lexically similar in both ontologies. When aligning DBpedia and GeoNames however, we saw that the values in GeoNames were abstract codes and it was not the case that they were lexically similar to the name of the corresponding class in DBpedia.

The extensional approach described by Parundekar is similar to Bayes-ReCCE in that it searches through possible restriction classes and uses sets of matched instances to find the support for these features describing a complex correspondence. It is capable of finding CAV correspondences, but in addition it can find highly complicated correspondences with multiple restriction features. It does not consider CAT or CAE correspondences. When designing Bayes-ReCCE we made a deliberate decision not to consider more complicated correspondences than CAV with a single restriction feature, as we did not want to produce correspondences that would be difficult to describe to a user, and would be difficult for them to evaluate. In addition the ability to have multiple restriction features in a correspondence means that there is a higher chance that the correspondences we detect will be over fitting the data, though it is difficult to test if this is a serious problem. In the evaluation of their HTS approach matching DBpedia and GeoNames, Parundekar et al. found 16 complex correspondences, all of which belonged to the CAT pattern. They did not find any complex correspondences with multiple restriction features. Out of the 16 correspondences they detected, 9 were correct compared to the gold standard used in our evaluation. This would give them a precision score of 0.56, a recall of 0.3 and an F1 score of 0.39.

In order to undertake some comparison with the Parundekar approach, our evaluation was concerned with Class by Attribute Value correspondences. Correspondences of this form consist of a Restriction Class – defined by an attribute and value pair – in one ontology, a named class in a second ontology and a relationship between the classes, usually equivalence or subsumption.

As GeoNames only provides a single class geo:Feature, all CAV correspondences between GeoNames and DBpedia can be expected to consist of a restriction class in GeoNames and a named class in DBpedia. Furthermore as geo:Feature was assumed to be broadly equivalent to dbpedia-owl:Place, this evaluation assumed that all named DBpedia classes in the correspondences will be sub classes of dbpedia-owl:Place. The procedure for our evaluation was to first find all CAV correspondences fitting the description above in the set of published correspondences which were discovered using HTS. Then, for each sub class of dbpedia-owl:Place, the Bayes-ReCCE method was used to discover if there was an attribute-value pair in GeoNames which could specify a CAV correspondence between a restriction class in GeoNames and the given sub-class. Twenty instances of each sub-class of dbpedia-owl:Place were used to train the Bayes-ReCCE detector. This number was selected as in the evaluation as it was previously discovered through experimental approaches that no significant improvement in performance was shown after 20 instances. The results of using Bayes-ReCCE are then compared with the CAV correspondences found using HTS. The most important metric is the number of valid correspondences each approach found. In addition, in the case where both methods find a CAV correspondence to a given DBpedia class, a record is made of which method produces a restriction class that more closely resembles the DBpedia class.

Having undertaken the evaluation, the results show that both methods produced similar numbers of acceptable correspondences between GeoNames and DBpedia with each producing 12. Bayes-ReCCE (see Table 2) however, produced only one incorrect correspondence:

- *Stadium* ≡ *featureCode* = *L.PRK*

While the HTS method produced four incorrect correspondences:

- *HistoricPlace* ≡ *inCountry* = *US*
- *Stadium* ≡ *parentFeature* = *2635167*
- *ProtectedArea* ≡ *inCountry* = *CA*
- *Building* ≡ *inCountry* = *GB*

This gives a false positive rate of 0.25 for the HTS method versus 0.08 for SP2.

The nature of these incorrect correspondences is interesting, in that for HTS, the attribute-value pairs used in the restriction class for the correspondence refer to the location of the feature and are very clearly inappropriate - being located in the country Great Britain is very much not a defining characteristic of being a building for example. The mistake produced by Bayes-ReCCE is much less obvious. Stadia are often referred to as parks, it is conceivable that the training data contained many examples of a stadium that were mistakenly labelled as a park.

Table 2. Class by Attribute Value correspondences between DBpedia and Geonames, discovered using the Bayes-ReCCE detection method.

Named Class	**Restriction Class**	Restriction Meaning
Park	featureCode = L.PRK	Park: an area, often of forested land, maintained as a place of beauty, or for recreation
ProtectedArea	featureCode = L.PRK	Park:
Mountain	featureCode = T	Mountain, hill, rock, ...
BodyOfWater	featureClass = H	Stream, lake, ...
Country	featureCode = A.PCLI	Independent political entity
Island	featureCode = T.ISL	Island: a tract of land, smaller than a continent, surrounded by water at high water
Airport	featureCode = S.AIRP	Airport: a place where aircraft regularly land and take off, with runways, navigational aids, and major facilities for the commercial handling of passengers and cargo
Hospital	featureCode = S.HSP	Hospital: a building in which sick or injured, especially those confined to bed, are medically treated
Stadium	featureCode = L.PRK	Park:
Skyscraper	featureCode = S.BLDG	Building: a structure built for permanent use, as a house, factory, etc.
Bridge	featureCode = S.BDG	Bridge: a structure erected across an obstacle such as a stream, road, etc., in order to carry roads, railroads, and pedestrians across
Lake	featureClass = H.LK	A large inland body of standing water
River	featureCode = H.STM	A body of running water moving to a lower level in a channel on land

There were three cases where HTS produced correspondences that, while not incorrect, could be improved on. These were:

- *Island* ≡ *featureClass=T*
- *Hospital* ≡ *featureClass=S*
- *Skyscraper* ≡ *featureClass=S*

For all of these cases, SP2 produced the more accurate correspondences

- *Island* ≡ *featureCode = T.ISL*
- *Hospital* ≡ *featureCode = S.HSP*
- *Skyscraper* ≡ *featureCode = S.BDG*

There were no cases where HTS produced correspondences that were more specific than those produced by Bayes-ReCCE .

Bayes-ReCCE was shown to be more capable of finding correspondences for classes with very few instances. For example the class Park with 21 instances, which Bayes-ReCCE produced a correct correspondence, and HTS produced no correspondence. Also, two out of the three cases where Bayes-ReCCE produced a more precisely defined correspondence were for classes with few instances – Hospital which had 23 instances in the training set and Skyscraper which had 36.

One further observation that should be noted, is that while the HTS detection method is capable of finding complex correspondences with restriction classes defined by multiple properties, no correspondences of this form were found between GeoNames restriction classes and named DBpedia classes. Bayes-ReCCE is only capable of detecting correspondences between restriction classes defined by a single attribute, but this was sufficient for this task.

In summary this evaluation shows that Bayes-ReCCE and HTS were both capable of finding the same number of valid CAV correspondences between DBpedia and GeoNames – 12 each. In one quarter of these cases however, Bayes-ReCCE produced more accurate complex correspondences – that is ones where the restriction class in the correspondence produced by Bayes-ReCCE was more closely related to the named class in the correspondence. It would also appear that Bayes-ReCCE performed better in relation to HTS when there were less training instances available.

One of the main limitations of this evaluation is that with only 14 CAV correspondences in total found between the two search methods, it is difficult to say with great certainty that Bayes-ReCCE will generally find more accurate CAV correspondences than HTS. A further limitation of this evaluation is that the training sets used for the Bayes-ReCCE detection method were drawn at random, and it is possible that different sampling could produce different results. This is especially true in the case where only a small number of matched instances are available – it is very possible that there is a systematic reason why those particular instances have been matched,

which in turn will mean that the available instances are not representative of the population as a whole. The goal of the evaluation was to show that Bayes-ReCCE can outperform HTS while requiring less training data. It is difficult to conclude strongly that this is the case given the evidence provided, but it is clear that both methods produced very similar results.

It **is** relatively safe to conclude from this evaluation at least there was nothing gained by using much a larger set of training data. This in itself is an important result. Often when performing ontology mapping the number of matched instances available will be relatively low, and in such cases, by using the Bayes-ReCCE method it should be possible to begin searching for correspondences between the taxonomic portions of the ontologies, without first searching for more matches between the instances of the ontologies. Furthermore, using smaller training sets places less strain on available storage and computing resources.

8. Related Work

As schema matching can be considered to be closely related to ontology matching [26], it is worth exploring the database community for work that attempts to identify complex correspondences. The Infosphere Data Architect [27] is a leading commercial tool that aids users in data mapping by: first detecting automatically one-to-one matches between elements in the schema; then displaying visually these relationships so that the user can add to or remove from this set; then the user has the option to add expressions, such as filter or join operations to the mappings. The mappings combined with the additional expressions are then used to automatically generate scripts which can transfer data from one schema representation to the other. The key challenge with complex matches is that the space of possible matching candidates is possibly unbounded, and evaluating every candidate becomes hard. Incorporating a novel explanation mechanism, iMAP [28] uses two main techniques to search the space effectively, employing a set of specialized searchers and aggressively using various types of domain knowledge to guide the search.The Tupelo framework [29] is a more generalised analysis of the data mapping problem that takes into account the need to find transformation expressions. This framework provides a calculus for describing the common recurring problems that are encountered when mapping data and lists the solutions that can be used. An important aspect of Tupelo is that it emphasises treating mapping discovery as an *example-driven* search in a space of transformations. This approach allows it to generate queries encompassing a wide range of structural and semantic heterogeneities encountered in relational data mapping.

Research to aid the discovery of complex correspondences for semantic web and linked data communities has also been maturing. The COMA Match system was first designed as a framework for combining schema matching approaches and was subsequently extended to match ontologies. The COMA toolset has seen continual development, and as of 2011 includes the Enhancement Engine component [30] which can take the one to one mappings produced by the schema matching approaches and use these to create complex correspondences. The search

strategy it uses is to first examine possible correspondences involving sets of elements which are close together in the ontology graphs, to determine if any transformation correspondences can be found. It then uses external linguistic oracles to discover additional correspondences using the element names. Qin et al. [31] has proposed the use of multi relational data mining techniques to detect complex correspondences between ontologies. As was the case with InfoSphere Data Architect, the process begins by generating one to one matches between the classes and between relations in the ontologies. Object reconciliation is then used to find common instances in the ontologies. The system seeks to find as many shared instances as possible. In addition, there is a feedback loop where information about the shared instances discovered at this stage are used to help find more matches between the classes and relations, which in turn are used to find more shared instances. The approach described in [5] uses correspondence patterns to detect complex correspondences. It analyses the structure of the ontologies and uses a series of string manipulation operations to test if the words used in the names classes and properties in one ontology can be re-arranged to form names similar to the ones used in the second ontology. The advantage of this approach is that it does not require any matches instances in order to be able to detect the complex correspondences. Identifying complex (one-to-many or many-to-many) correspondences can be seen as closely related to the problem of determining the semantic relation type of correspondences. For example, a many-to-one situation where several concepts of the first ontology are related to the same concept of the second ontology can indicate is-a or part-of relationships between these concepts. In [32], the authors propose and evaluate a framework two-step enrichment approach to determine semantic ontology mappings that enhances existing match tools.

A rule-learning-based approach for detecting complex correspondences in Linked Data has been proposed in [33][34]. Derived from classical Inductive Logic Programming, the approach in [33] uses instance mappings as training data and employs tailoring heuristics to improve the learning efficiency, with initial evaluation showing that the generated Horn-rule mappings are meaningful. In [33], they demonstrate that learning and refinement of non-trivial ontology alignments can be achieved from instances through utilization of inductive rule learning algorithms, by reformulating as problems of association rule mining and separate-and-conquer rule learning, respectively. A novel approach has been proposed to identify complex correspondences based on a the unified top-k match graph for top-k matchings [34]. This uses a clustering problem to group attributes that show ambiguity and are closely related, and for these groups, quality is assessed and, if appropriate, complex correspondences are derived.

9 Conclusions

In the beginning of this article we highlighted the lack of homogeneity between classes used in Linked Open Data. We noted that this heterogeneity means that describing the relationship between classes in ontologies is not always possible using elementary links, and that often complex correspondences are needed. Many complex correspondences can be described as

commonly re-occurring patterns, containing a small number of features. Our algorithm, Bayes-ReCCE, automates the process of finding complex correspondences which fit these patterns by searching for these features in matched instance data.

In order to support this search, we proposed a probabilistic metric which estimates the similarity of the extensional sets of the classes in a correspondence. This measure is based on the Jaccard Index, but whereas the Jaccard Index requires a closed world assumption, our similarity measure makes an open world assumption. One of the weaknesses of the Jaccard is that it will draw strong conclusions when small amounts of data are available, where as our measure is probabilistic and small amounts of data will generate lower probabilities. In addition the measure we employ allows us to define rules for our prior beliefs about the similarity of the sets.

We demonstrated that complex correspondences occur in the DBpedia, LinkedMDB and GeoNames knowledge bases, and that Bayes-ReCCE was capable of detecting these correspondences automatically. This is significant, as these are large, real-world datasets, and it is difficult even for a human to understand the relationships between them. For example, we found that even the published list of correspondences between GeoNames and DBpedia was incomplete, missing 11 class by attribute value correspondences compared to the set of 32 correspondences already published by Geonames.

As an extensional approach, Bayes-ReCCE, depends on the presence of (1) equivalent instance data in the ontologies or linked data-sets to be matched and (2) instance equivalence interlines. This limits its applicability to ontologies or linked data where such data exists but since this is a probabilistic tool designed deal with the reality of large, public, linked data repositories the restriction of having data instances is not often going to be an issue in practice. Since any mapping or alignment activity requires some commonality of the domains of interest we can also assume for many cases there will be common instances, but this is worth considering before deploying the method. Since the method requires some set of matched instances as input it is our working assumption that this data will be provided by human supervision of the process, perhaps as part of a wider alignment/mapping tool-chain and we make no claim that the method is suitable for totally unsupervised application to non-overlapping datasets. Intensional approaches in theory have wider applicability but when the details of their current algorithms are investigated, e.g. [5], we see serious limitations such as an inability to deal with attributes other than Boolean values whereas Bayes-ReCCE deals with all attribute domains.

We demonstrated that Bayes-ReCCE performs well with even small sets of matched data. When matching DBpedia and GeoNames a sample of only 5 matched instances per target class allowed for a mean F1 score between 0.58 and 0.62. This is significant as matching instance data is not in general a trivial task. DBpedia, GeoNames and LinkedMDB are well established datasets in the LOD cloud with many interlinks. When adding a new dataset to the cloud, instance links may be at a premium, however Bayes-ReCCE should still be able to produce some meaningful results in situations like this.

We also observed that the extensional set similarity between the classes in published correspondences can be quite low, with the mean Jaccard index of correspondences from GeoNames to DBpedia being only 0.43. The predicate *owl:equivalentClass* is not always the most appropriate for describing the relationship between corresponding classes. Alternative predicates such as *skos:closeMatch* can be ambiguous and may not provide sufficient information to allow the correspondence to be used by a third party. We illustrated that providing a summary of the information used to initially select the correspondence could allow a third party to draw their own conclusions on the nature of the relationship being described, and make a more informed decision on its use.

Of course a limitation to the solution is that it heavily relies on the availability of a Knowledge Base's TBox. Whilst it is good practice to have such schema, in reality it may not always be possible to have such schemas available.

Bayes-ReCCE advances the state of the art in that it has been demonstrated to be effective when using in the order of tens of instances per class, while other approaches use in the order of thousands. In addition Bayes-ReCCE explains its output in terms of correspondence patterns which can help users interpret the results. This is valuable as interpreting results is seen as one of the leading challenges in ontology mapping.

Acknowledgements

This research is supported by Science Foundation Ireland through the FAME Strategic Research Cluster (Grant 08/SRC/I1403) and partially through the CNGL Programme (Grant 12/CE/I2267) in the ADAPT Centre (www.adaptcentre.ie) at Trinity College Dublin and this work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644055 (ALIGNED, http://www.aligned-project.eu)

.

References

[1] C. Bizer, "Linked data-the story so far," *Int. J. Semant. Web Inf. Syst.*, vol. 4, no. 2, pp. 1–22, Jan. 2009.

[2] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A large ontology from wikipedia and wordnet," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 6, no. 3, pp. 203–217, Sep. 2008.

[3] G. A. Miller, "WordNet : A Lexical Database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[4] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia - A crystallization point for the Web of Data," *Web Semant. Sci. Serv. Agents World Wide Web*, no. 7, pp. 154–165, Sep. 2009.

[5] D. Ritze, C. Meilicke, O. Sváb-Zamazal, and H. Stuckenschmidt, "A pattern-based ontology matching approach for detecting complex correspondences," in *Proc. of Int. Workshop on Ontology Matching (OM)*, 2009.

[6] F. Scharffe, "Correspondence patterns representation. PhD thesis," University of Innsbruck, 2009.

[7] R. Parundekar, C. A. Knoblock, and L. Ambite, "Linking and Building Ontologies of Linked Data," *9th International Semantic Web Conference*, 2010. .

[8] R. Parundekar, C. A. Knoblock, and L. Ambite, "Discovering Concept Coverings in Ontologies of Linked Data Sources," in *11th International Semantic Web Conference*, 2012, pp. 427–443.

[9] M. Wick, "Geonames," 2006. [Online]. Available: www.geonames.org. [Accessed: 21-May-2015].

[10] O. Hassanzadeh and M. Consens, "Linked movie data base," in *Linked Data on the Web (CEUR workshop Proceedings)*, 2009, vol. 538.

[11] Gamma, Helm, Johnson, and Vlissides, *Design Patterns*. Addison-Wesley, 1994.

[12] J. David, J. Euzenat, F. Scharffe, and C. Trojahn dos Santos, "The alignment API 4.0," *Semant. Web*, vol. 2, no. 1, pp. 3–10, 2011.

[13] P. Shvaiko and J. Euzenat, "Ontology matching: state of the art and future challenges," *IEEE Trans. Knowl. Data Eng.*, vol. X, no. 1, pp. 1–20, 2012.

[14] G. Correndo and N. Shadbolt, "Translating expressive ontology mappings into rewriting rules to implement query rewriting," in *International Workshop on Ontology Matching*, 2011.

[15] S. Hellmann, J. Lehmann, and S. Auer, "Learning of OWL class descriptions on very large knowledge bases," *Int. J. Semant. Web Inf. Syst.*, vol. 5, no. 2, 2009.

[16] B. De Finetti, "Probabilism - A critical essay on the theory of probability and on the value of science," *Erkenntnis*, vol. 31, no. 2, pp. 169–223, 1989.

[17] C. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. Springer, 2006, pp. 113 – 117.

[18] J. K. Shultis and N. D. Eckhoff, “Selection of Beta Prior Distribution Parameters from Component Failure Data,” *IEEE Trans. Power Appar. Syst.*, vol. PAS-98, no. 6, pp. 400–407, 1979.

[19] J. Li, J. Tang, Y. Li, and Q. Luo, “RiMOM: A dynamic multistrategy ontology alignment framework,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 8, pp. 1218–1232, 2009.

[20] W. Hu, Y. Qu, and G. Cheng, “Matching large ontologies: A divide-and-conquer approach,” *Data Knowl. Eng.*, vol. 67, pp. 140–160, 2008.

[21] P. Lambrix and H. Tan, “SAMBO-A system for aligning and merging biomedical ontologies,” *Web Semant.*, vol. 4, pp. 196–206, 2006.

[22] M. H. Seddiqui and M. Aono, “An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size,” *J. Web Semant.*, vol. 7, pp. 344–356, 2009.

[23] H. Halpin, P. J. Hayes, J. P. Mccusker, D. L. Mcguinness, and H. S. Thompson, “When owl : sameAs isn’t the Same : An Analysis of Identity in Linked Data,” in *ISWC*, 2011.

[24] P. Shvaiko and J. Euzenat, “Ten challenges for ontology matching,” *Move to Meaningful Internet Syst. OTM 2008*, vol. 5332/2008, pp. 1164–1182, 2008.

[25] M. Schmachtenberg, C. Bizer, H. Paulheim, "Adoption of the Linked Data Best Practices in Different Topical Domains" The Semantic Web -- ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, pp 245-260, 2014.

[26] P. Shvaiko and J. Euzenat, “A Survey of Schema-Based Matching Approaches,” *J. Data Semant.*, vol. 4, pp. 146 – 171, 2005.

[27] E. Wilson, S. Vibhute, C. Bhatia, R. Jain, L. Perniu, S. Raveendramurthy, and R. Samuel, *Getting Started with InfoSphere Data Architecht*, 1st ed. IBM, 2011.

[28] R. Dhamankar, Y. Lee, A. Doan, A. Halevy, P. Domingos, "Discovering complex semantic matches between database schemas", Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, ACM, 2004.

[29] G. H. L. Fletcher and C. M. Wyss, “Towards a General Framework for Effective Solutions to the Data Mapping Problem,” *Lect. Notes Comput. Sci.*, vol. 5880, pp. 37–73, 2009.

[30] S. Massmann, S. Raunich, and D. Aumüller, “Evolution of the coma match system,” in *International Workshop on Ontology Matching*, 2011.

[31] H. Qin, D. Dou, and P. Lependu, “Discovering Executable Semantic Mappings Between Ontologies,” in *2007 OTM Confederated international conference on On the move to meaningful internet systems: CoopIS, DOA, ODBASE, GADA, and IS*, 2007, pp. 832–849.

[32] P. Arnold, E. Rahm, "Enriching ontology mappings with semantic relations", Data & Knowledge Engineering, Volume 93, September 2014, Pages 1-18

[33] W. Hu, J. Chen, H. Zhang, Y. Qu, "Learning Complex Mappings between Ontologies", The Semantic Web: Joint International Semantic Technology Conference, JIST 2011, Hangzhou, China, Volume 7185, Lecture Notes in Computer Science, pp 350-357, 2011.

[33] F. Janssen, F. Fallahi, J. Noessner, H. Paulheim, "Towards Rule Learning Approaches to Instance-based Ontology Matching", 1st International Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data (Know@LOD), pp 13-18, 2012.

[34] A. Gal, T. Sagi, M. Weidlich, E. Levy, V. Shafran, Z. Miklós, N. Quoc Viet Hung, "Making sense of top-k matchings: a unified match graph for schema matching", Ninth International Workshop on Information Integration on the Web (IIWeb '12), ACM, 2012.

[35] A. Crotti, B. Walshe, D. O'Sullivan, "Enhanced faceted browsing of a WW1 dataset through ontology alignment", 17th International Conference on Information Integration and Web-based Applications & Services (iiWAS 2015), Brussels, Belgium, 11-13 December , ACM, pp70 - 77, 2015.

[36] H. Paulheim, C. Bizer, "Improving the Quality of Linked Data Using Statistical Distributions", International Journal on Semantic Web and Information Systems (IJSWIS), Vol. 10, Issue 2, 2014.

[37] J. Lehmann, S. Auer, L Bühmann, S. Tramp, "Class expression learning for ontology engineering", Web Semantics: Science, Services and Agents on the World Wide Web, Volume 9, Issue 1, pp 71-81, 2011.

[38] Johanna Völker, Mathias Niepert, "Statistical Schema Induction", The Semantic Web: Research and Applications, Lecture Notes in Computer Science, Vol. 6643, pp 124-138, Springer.

[39] P. Gillet, C. Trojahn, O. Haemmerlé, C. Pradel, "Complex Correspondences for Query Patterns Rewriting", OM'13 8th International Conference on Ontology Matching, Vol. 1111, pp 49-60, CEUR-WS, 2013.

[40] G. De Melo, F. Suchanek and A. Pease, "Integrating YAGO into the Suggested Upper Merged Ontology, "m 2008 20th IEEE Int. Conf. Tools with Artif,. Intell., pp . 190-193, Nov. 2008