Terminological Cluster Trees for Disjointness Axiom Discovery

Giuseppe Rizzo(✉), Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito

LACAM – Dipartimento di Informatica,
Università degli Studi di Bari "Aldo Moro", Bari, Italy
{giuseppe.rizzo1,
claudia.damato,nicola.fanizzi,floriana.esposito}@uniba.it
http://lacam.di.uniba.it

Abstract. Despite the benefits deriving from explicitly modeling concept disjointness to increase the quality of the ontologies, the number of disjointness axioms in vocabularies for the *Web of Data* is still limited, thus risking to leave important constraints underspecified. Automated methods for discovering these axioms may represent a powerful modeling tool for knowledge engineers. For the purpose, we propose a machine learning solution that combines (unsupervised) distance-based clustering and the divide-and-conquer strategy. The resulting *terminological cluster trees* can be used to detect candidate disjointness axioms from emerging concept descriptions. A comparative empirical evaluation on different types of ontologies shows the feasibility and the effectiveness of the proposed solution that may be regarded as complementary to the current methods which require supervision or consider atomic concepts only.

1 Introduction

With the growth of the Web of Data, along with the *Linked Data* initiative [13], a large number of datasets are being published using a standard data model that connects lots of knowledge bases within a uniform semantic space. In this context, Web ontologies are used as formal vocabularies that support many important services based on *automated reasoning*, such as *classification*, *query answering*, *population* and *enrichment*, or *reconciliation* (*instance matching*). In this perspective, ontologies represent a means to ensure the quality of data.

Debugging strategies may be employed to prevent the introduction of conflicting assertions that hinder the employment of reasoning services. However many ontologies still represent simplified data models for the targeted domain failing to capture some underlying intended constraints [21]. A common problem is the lack of an explicit representation of negative knowledge, usually expressed in the form of *disjointness axioms*. Conversely, the acquisition of such axioms may enhance the mentioned rich services.

In the literature, sundry approaches for discovering disjointness axioms have been proposed. Recent methods apply association rule mining [18,19]. However,

E. Blomqvist et al. (Eds.): ESWC 2017, Part I, LNCS 10249, pp. 184–201, 2017.
DOI: 10.1007/978-3-319-58068-5_12

they can capitalize on the available of intensional knowledge only to some marginal extent. In these works, heterogeneous sources are exploited, with most of the features being lexical, also based on external corpora. Additionally, most of the approaches move from the assumption that disjointness may hold among concepts when the sets of their instances, that can be thought of as empirical approximations of their extensions, do not tend to *overlap* [11,20]. Hence a more data-driven approach could be exploited: it may derive from finding partitions of similar individuals occurring in the knowledge base according to some criterion of choice, by maximizing the separation (i.e. minimizing the overlap) among different partitions. This objective boils down to a *clustering problem* [1] which is a classic topic in *machine learning*. In the context of the *Semantic Web* (SW), clustering methods for individuals described by ontologies have been proposed, extending classic algorithms such as K-MEANS or K-MEDOIDS [9] with the ability of taking into account intensional knowledge. They have been mainly employed for concept learning or for the automated detection of *concept drift* or *novelty* [10].

In the line of the unsupervised statistical approaches, ours relies on methods that produce hierarchical clustering structures, while taking into account the intensional knowledge provided by the ontology. The goal is to derive potential disjointness axioms, by exploiting the background knowledge on the schema, similarly to *relational learning* frameworks [15]. Indeed, we adopt an approach based on *conceptual clustering*, that aims at learning intensional descriptions of emerging clusters of individuals that may involve even complex concept descriptions, differently from other unsupervised approaches. Our solution is based on a novel form of logical tree model [7], dubbed *terminological cluster tree* [16], that can be regarded as an extension of the terminological decision tree [8]. Both are produced through *divide-and-conquer* algorithms, but while the latter are essentially classifiers that solve supervised concept learning problems exploiting information-based heuristics (*information gain* or other *purity* measures), the former rely on specific metrics and on a notion of *cluster prototype* [9,10].

Unlike other (unsupervised) clustering models, the proposed solution also aims at intensional definitions of the clusters, i.e. concept descriptions that describe their individuals. Another advantage is that the number of clusters – which has a strong impact on the quality of the clustering structure – is not a required parameter; conversely, descends from the number of dense data regions found in the given instance space. Indeed, in the induction of (terminological) cluster trees this number depends on a notion of *purity*, that determines the stop condition for further branchings to prevent compromising the clusters *separation*. Once the tree is grown, groups of (disjoint) clusters located at sibling nodes identify concepts involved in candidate disjointness axioms to be derived. The discovered axioms can be validated by a domain expert/ontology engineer and/or may even be automatically involved in a debugging process for eliciting cases of inconsistency which cannot arise when disjointness axioms are lacking.

Summary. In the next section, related works are briefly surveyed. In Sect. 3, disjointness axiom discovery problem is formalized as a clustering problem for individuals in a knowledge base. Section 4 illustrates the approach to the induction of terminological cluster trees and their application to the targeted problem. Section 5 presents a comparative experimental evaluation of the proposed solution on common ontologies. Finally, Sect. 6 concludes this work delineating future research directions.

2 Related Work

The problem of discovering the disjointness axioms to enrich and improve the quality of ontological knowledge bases has been receiving a growing attention. In early works, the mentioned *strong disjointness assumption* [5] (SDA), which states that the children of a common parent in the subsumption hierarchy should be disjoint, has been exploited in a pinpointing algorithm for semantic *clarification* (i.e. the process of automatically enriching ontologies by appropriate disjointness statements [17]. Focusing first on text and successively on RDF datasets, unsupervised methods for mining axioms, including disjointness axioms, have been proposed [12,20]. Their main limitation is the inability to exploit background knowledge, which on the contrary may help in increasing the number of axioms discovered while filtering out unnecessary or wrong axioms. The main limitations of supervised methods is the necessity of axioms for training that may demands costly work by domain experts.

Besides, methods based on *relational learning* [15] and *formal concept analysis* [3] have been proposed, but none specifically aimed at assessing the quality of the induced axioms. This is pointed out also in [19] and additional approaches [11,18] based on *association rule mining* have been introduced to better address this limitation. The goal was studying the correlation between classes. Specifically, *(negative) association rules* and the use of a *correlation coefficient* have been considered. Also in these cases, background knowledge is not explicitly exploited. In [15], a tool for *repairing* various types of ontology modeling errors is described; it uses methods from the DL-Learner framework [14] to enrich ontologies with axioms induced from existing instances.

Our solution is based on an unsupervised approach, deriving from previous works on concept learning and inductive classification [6]. Specifically, we propose a hierarchical conceptual clustering method that is able to provide intensional cluster descriptions. It exploits a novel form of a family of semi-distances for the individuals in knowledge bases [6] which involves reasoning on the knowledge base. The method is grounded on the notion of medoid as cluster prototype to give a topological structure to the representation of the instance space [1]. Related, but partitive clustering approaches [1], such as the bisecting k-medoids [9] or the *partition around medoids*, combined with evolutionary programming [10] have been proposed. They cluster individuals in Web ontologies by exploiting metrics that are related to those adopted in this work. They can be easily extended for producing hierarchical structures of clusters. However, the derivation of related concepts as intensional definitions for the clusters requires the adoption of additional and suitable concept learning algorithms.

Specifically, the method proposed in this paper relies on logic tree models [4] which essentially adopt a divide-and-conquer strategy to derive a hierarchical structure. The learning method can work both in supervised and unsupervised mode, depending on the availability of information about the instance classification to be exploited for separating sub-groups of instances. Terminological decision trees were derived [8] in the former case, while for the latter case, first-order logic clustering trees [7] were proposed to induce concepts expressed in *clausal logics* for the clusters. The C0.5 system, which is integrated in the TILDE framework [4], is able to induce concepts as conjunctions of literals (clause bodies) installed at inner nodes. Almost all these exiting methods are grounded on the exploitation of an heuristic based on the *information gain*, employed in the supervised case. Differently, our approach tends to maximize the separation between cluster medoids according to a semi-distance measure that can be also computed before the learning phase [6] making a more efficient use of the computationally expensive reasoning services.

3 Disjointness Discovery as a Conceptual Clustering Problem

In this section, we formalize the problem of *discovering concept disjointness axioms* from an ontological knowledge base in terms of a *clustering* task. We will borrow notation and terminology from *Description Logics* (DLs) [2], being the theoretical foundation of the standard representation languages for the SW. Hence, we will use the terms *concept* (*description*) and *role* as synonyms of *class* and *property*, respectively and we will denote a *knowledge base* (KB) with $\mathcal{K} = \langle T, \mathcal{A} \rangle$, where $\mathcal{T}$ is the *TBox* (containing terminological axioms regarding concepts and roles) and $\mathcal{A}$ is the *ABox* (containing concept/role assertions regarding individuals). $\mathsf{Ind}(\mathcal{A})$ will denote the set of individuals (resource names) occurring in $\mathcal{A}$. *Subsumption*, *equivalence* and logic *entailment* will be denoted with the usual symbols.

Before formalizing the problem of discovering concept disjointness axioms, for the sake of clarity and completeness, we recall some basic *clustering* methods. Clustering is an unsupervised learning task aiming at grouping a collection of objects into subsets or *clusters*, such that those within each cluster are more closely related/similar to one another, than objects assigned to different clusters [1]. In the general setting, an object is usually described in terms of features from a selected set $\mathcal{F}$; a measure of similarity between objects is expressed in terms of a distance function, e.g. in the case of attribute-value datasets of objects that are often described by tuples of numeric features, the *Euclidean* distance (or its extensions) is typically adopted. A more complex goal is to move from flat to natural *hierarchical clustering* structures. Another difference among the various clustering models is related to the form of membership of the objects with respect to the clusters. In the simplest (*crisp*) case, e.g. K-MEANS, cluster membership *can be exclusive*: each object belongs to one cluster. Extensions, such as FUZZY C-MEANS or EM [1], admit overlapping clusters as the objects exhibit a graded membership (*responsibility*) w.r.t. the clusters. An interesting class of methods is represented by the *conceptual clustering* approaches. In the resulting clustering structures, the objects are arranged into clusters that are intentionally, rather than extensionally, defined. Differently from other methods, conceptual clustering algorithms may exploit available background knowledge for building descriptions for each cluster. Besides the propositional data representations (or the equivalent vector spaces mentioned above) more expressive data representation, e.g. through richer logic languages may be necessary. When such expressive representations are considered, suitable (dis)similarity measures have to be adopted.

Moving from the observation that a disjointness axiom, involving two or more concepts, may hold if their extensions do not overlap (as introduced in Sect. 1), the task of discovering disjointness axioms may be regarded as an unsupervised conceptual clustering problem aiming at finding separate partitions of individuals of the KB (such that each subset consists of similar individuals, according to a given similarity criterion) and producing intensional descriptions for them. The problem is defined as follows:

Definition 3.1 (Disjointness axiom discovery as a conceptual clustering problem)

Given
- *a knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$
- *a set of training individuals* $\mathbf{I} \subseteq \mathsf{Ind}(\mathcal{A})$

Find
- *a partition* Π *of* $\mathbf{I}$ *in a set of pairwise disjoint clusters* $\Pi = \{\mathbf{C}_1, \ldots, \mathbf{C}_{|\Pi|}\}$
- *for each* $i = 1, \ldots, |\Pi|$, *a concept description* D_i *that describes* $\mathbf{C}_i$, *so that:*
  $\forall a \in \mathbf{C}_i :\ \mathcal{K} \models D_i(a)$ *and*
  $\forall b \in \mathbf{C}_j, j \neq i :\ \mathcal{K} \models \neg D_i(b)$.
  Hence $\forall D_i, D_j, i \neq j :\ \mathcal{K} \models D_j \sqsubseteq \neg D_i$.

Note that the number of clusters (say $K = |\Pi|$) is not a required parameter to be provided tentatively. Also note that, the problem of discovering disjointness axioms resorting to machine learning methods can be formalized in different ways, depending on the type of approach (supervised or unsupervised) to be employed. In the next section, a solution to the formalized problem is presented.

4 Terminological Cluster Trees for Disjointness Learning

The proposed approach is grounded on a two-steps process. In the first step, given a knowledge base, clusters and the related concepts that describe them are discovered and organized in a tree structure. In the second step, the induced structure is exploited for learning a set of disjointness axioms. The model can be formally defined as follows:

Definition 4.1 (Terminological cluster tree). *Given a knowledge base* $\mathcal{K}$, *a* terminological cluster tree *(TCT) is a binary logical tree where each node stands for a cluster of individuals* $\mathbf{C}$ *and such that:*

Algorithm 1. Routines for inducing a TCT

```
const $\nu$, $\delta$: thresholds

function INDUCETCT($\mathbf{I}$, $C$): TCT
input $\mathbf{I}$: set of individuals
      $C$: concept description
begin
    $T \leftarrow$ new TCT
    if STOPCONDITION($\mathbf{I}$, $\nu$) then
        $T \leftarrow \langle \text{null}, \mathbf{I}, \text{null}, \text{null} \rangle$
    else
        $\mathbf{S} \leftarrow \rho(C)$  {set of candidate specializations}
        $E^* \leftarrow$ SELECTBESTCONCEPT($\mathbf{S}$, $\mathbf{I}$)
        $\langle \mathbf{I}_{\text{left}}, \mathbf{I}_{\text{right}} \rangle \leftarrow$ SPLIT($\mathbf{I}$, $E^*$)
        $T_{\text{left}} \leftarrow$ INDUCETCT($\mathbf{I}_{\text{left}}$, $E^*$)
        $T_{\text{right}} \leftarrow$ INDUCETCT($\mathbf{I}_{\text{right}}$, $\neg E^*$)
        $T \leftarrow \langle E^*, \mathbf{I}, T_{\text{left}}, T_{\text{right}} \rangle$
    return $T$
end

function SPLIT($\mathbf{I}$, $E$): pair of sets of individuals
input $\mathbf{I}$: set of individuals
      $E$: concept description
begin
    $\langle \mathbf{P}, \mathbf{N} \rangle \leftarrow$ RETRIEVEPOSNEG($\mathbf{I}$, $E$, $\delta$)
    $b \leftarrow p(\mathbf{P})$ { prototype }
    $c \leftarrow p(\mathbf{N})$ { prototype }
    $\mathbf{I}_{\text{left}} \leftarrow \emptyset$
    $\mathbf{I}_{\text{right}} \leftarrow \emptyset$
    for each $a \in \mathbf{I}$
        if $d(a, b) \leq d(a, c)$ then
            $\mathbf{I}_{\text{left}} \leftarrow \mathbf{I}_{\text{left}} \cup \{a\}$
        else
            $\mathbf{I}_{\text{right}} \leftarrow \mathbf{I}_{\text{right}} \cup \{a\}$

    return $\langle \mathbf{I}_{\text{left}}, \mathbf{I}_{\text{right}} \rangle$
end
```

- *each node contains a concept D (defined over the signature of $\mathcal{K}$) describing* $\mathbf{C}$*;*
- *each edge departing from an internal node corresponds to the outcome of the membership test of individuals with respect to D.*

A tree-node is represented by a quadruple $\langle D, \mathbf{C}, T_{left}, T_{right} \rangle$ with the left and right subtrees connected by either departing edge.

The construction of the model combines elements of logical decision trees induction (recursive partitioning and refinement operators for specializing concept descriptions) with elements of *instance-based learning* (a distance measure over the instance space). The details of the algorithms for growing a TCT (*step 1.*) and deriving intensional definitions of disjoint concepts (*step 2.*) are reported in the sequel.

4.1 Growing Terminological Cluster Trees

A TCT T is induced by means of a recursive strategy (see Algorithm 1), which follows the schema proposed for terminological decision trees (TDTs) [8]. The ultimate goal is to find a partition of *pure* clusters.

The main routine INDUCETCT is to be invoked passing $\mathbf{I}$ and $\top$ as parameters. In this recursive function, the base case tests the STOPCONDITION predicate, i.e. whether the measure of cohesion of the cluster $\mathbf{I}$ exceeds a given threshold ν. Further details about the heuristics and the stop condition will be reported later on.

In the inductive step, which occurs when the stop condition does not hold, the current (parent) concept description C has to be specialized using a refinement operator (ρ) that spans over a search space of concepts subsumed by C. A set of candidate specializations $\mathbf{S} = \rho(C)$ is obtained. For each $E \in \mathbf{S}$, the sets of positive and negative individuals, i.e. the instances of E and of $\neg E$, respectively denoted by $\mathbf{P}$ and $\mathbf{N}$, are retrieved by RETRIEVEPOSNEG. A tricky situation may occur when either $\mathbf{N}$ or $\mathbf{P}$ is empty for a given E (e.g. in the absence of disjointness axioms). In such a case, RETRIEVEPOSNEG assigns individuals in $\mathbf{I} \setminus \mathbf{P}$ to $\mathbf{N}$ (resp. in $\mathbf{I} \setminus \mathbf{N}$ to $\mathbf{P}$) when the distance between them and the prototype of $\mathbf{P}$ (resp. $\mathbf{N}$) exceeds the threshold δ. A representative element for $\mathbf{P}$ and $\mathbf{N}$ is determined as a prototype, i.e. their *medoid*, a central element with the minimal average distance w.r.t. the other elements in the cluster. Then, function SELECTBESTCONCEPT evaluates the candidate specializations in terms of the *cluster separation* computed through a heuristic (see Eq. 1) and returns the best concept $E^* \in \mathbf{S}$, that is the one for which the distance between the medoids of the related positive and negative sets is maximized. Then E^* is installed in the current node. Hence, the individuals are partitioned by SPLIT to be routed along the left or right branch of E^*. Differently from TDTs, the routine does not decide the branch where the individuals will be sorted according to a concept membership test (instance check): it splits individuals according to the distance w.r.t. the tow prototypes, i.e. the medoids of $\mathbf{P}$ and $\mathbf{N}$.

This divide-and-conquer algorithm is applied recursively until the instances routed to a node satisfy the stop condition. Note that, the number of the clusters is not required as an input but it depends on the number of branches grown: it is naturally determined by the algorithm according to the data distribution in the regions of the instance space.

The proposed approach relies on a downward refinement operator that can generate the concepts to be installed in child-nodes performing a specialization process on the concept, say C, installed in a parent-node or its complement:

ρ_1 by adding a concept atom (or its complement) as a conjunct: $C' = C \sqcap (\neg)A$;
ρ_2 by adding a general existential restriction (or its complement) as a conjunct: $C' = C \sqcap (\neg)(\exists)R.\top$;
ρ_3 by adding a general universal restriction (or its complement) as a conjunct: $C' = C \sqcap (\neg)(\forall)R.\top$;
ρ_4 by replacing a sub-description C_i in the scope of an existential restriction in C with one of its refinements: $\exists R.C_i' \in \rho(\exists R.C_i) \wedge C_i' \in \rho(C_i)$;
ρ_5 by replacing a sub-description C_i in the scope of a universal restriction with one of its refinements: $\forall R.C_i' \in \rho(\forall R.C_i) \wedge C_i' \in \rho(C_i)$.

Note that the cases of ρ_4 and ρ_5 are recursive. Please, also note that the refinement operator take the KB (and particularly the TBox) strictly into account. Indeed refinements that are consistent with the KB are always returned.

The algorithms for growing TCTs and TDTs share a common structure but differ on the criterion for selecting the test concepts installed in the nodes: while *information gain* is adopted by the latter, the procedure for TCTs resorts to a measure of distance defined over the individuals occurring in the knowledge base. Specifically, the heuristic for selecting the best refinement of the parent concept is defined as follows:

$$E^* = \underset{D \in \rho(C)}{\operatorname{argmax}}\, d\left(p(\mathbf{P}), p(\mathbf{N})\right) \tag{1}$$

where $\mathbf{P}$ and $\mathbf{N}$ are sub-clusters output by $\textsc{retrievePosNeg}(\mathbf{I}, D, \delta)$, $d(\cdot,\cdot)$ is a distance measure between individuals and $p(\cdot)$ is a function that maps a cluster to its *prototype*. As previously mentioned, the adopted $p(\cdot)$ computes the *medoid* of the cluster.

The required measure for individuals should capture aspects of their semantics in the context of the KB. We resort to variation of a language-independent dissimilarity measure proposed in previous works [6,9,10]. Given the knowledge base $\mathcal{K}$, the idea is to compare the behavior of the individuals w.r.t. a set of concepts $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ that is dubbed *context* or *committee* of features. For each $C_i \in \mathcal{C}$, a *projection function* $\pi_i : \mathsf{Ind}(\mathcal{A}) \to [0,1]$ is defined as a simple mapping:

$$\forall\, a \in \mathsf{Ind}(\mathcal{A}) \qquad \pi_i(a) = \begin{cases} 1 & \text{if } \mathcal{K} \models C_i(a) \\ 0 & \text{if } \mathcal{K} \models \neg C_i(a) \\ 0.5 & \text{otherwise} \end{cases} \tag{2}$$

where the third value (0.5) represents a case of maximal uncertainty on the membership. As an alternative, the estimate of the likelihood for a generic individual a of being an instance of C_i could be considered. Especially with densely populated ontologies (as those forming the Web of Data) the probability $\Pr[\mathcal{K} \models C_i(a)]$ may be estimated by $|r_{\mathcal{K}}(C_i)|/|\mathsf{Ind}(\mathcal{A})|$, where $r_{\mathcal{K}}()$ denotes the *retrieval* of a concept w.r.t. $\mathcal{K}$, i.e. the set of individuals of $\mathsf{Ind}(\mathcal{A})$ that (can be proven to) belong to C_i [2].

Hence, a family of distance measures $\{d_n^{\mathcal{C}}\}_{n \in \mathbb{N}}$ can be defined as follows: $d_n^{\mathcal{C}} : \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \to [0,1]$ with

$$d_n^{\mathcal{C}}(a,b) = \left[\sum_{i=1}^{m} w_i \left[1 - \pi_i(a)\pi_i(b)\right]^n \right]^{1/n} \tag{3}$$

Non uniform values for $\boldsymbol{w}$ can be considered to reflect the specific importance of each feature. For example it may be set according to an entropic measure [6,10] based on the average information brought by each concept:

$$\forall i \in \{1, \ldots, m\} \qquad w_i = - \sum_{k \in \{-1,0,+1\}} \mu_i(k) \log \mu_i(k) \tag{4}$$

where, given a generic $a \in \mathsf{Ind}(\mathcal{A})$, the following estimates can be used: $\mu_i(+1) \approx \Pr[\mathcal{K} \models C_i(a)]$, $\mu_i(-1) \approx \Pr[\mathcal{K} \models \neg C_i(a)]$ and $\mu_i(0) = 1 - \mu_i(+1) - \mu_i(-1)$.

The growth of a TCT can be stopped by resorting to a heuristic that is similar to the one employed for selecting the best concept description. This requires the employment of a threshold $\nu \in [0,1]$ for the value of $d(\cdot,\cdot)$. If the value is lower than the threshold, the branch growth is stopped.

4.2 Extracting Candidate Disjointness Axioms from TCTs

The procedure for discovering/extracting disjointness axioms requires a TCT as input. It is reported in Algorithm 2.

Algorithm 2. Routines for deriving disjointness axioms from a TCT

```
function deriveCandidateAxioms($T$): set of axioms
input $T$: TCT
begin
    $\mathbf{A} \leftarrow \emptyset$
    $\mathbf{CS} \leftarrow$ collect($\top, T$)
    for each $C \in \mathbf{CS}$ do
        for each $D \in \mathbf{CS}$ do
            if $(D \sqsubseteq \neg C) \notin \mathbf{A}$ and $C \not\equiv D$ then
                $\mathbf{A} \leftarrow \mathbf{A} \cup \{D \sqsubseteq \neg C\}$
    return $\mathbf{A}$
end
```

```
function collect($C, T$): set of concepts
input $C$: concept description
      $T$: TCT
begin
    let $T = \langle D, \mathbf{I}, T_{\text{left}}, T_{\text{right}} \rangle$
    if $T_{\text{left}} = T_{\text{right}} =$ null then        { leaf }
        return $\{C\}$
    else
        $\mathbf{CS}_{\text{left}} \leftarrow$ collect($C \sqcap D, T_{\text{left}}$)
        $\mathbf{CS}_{\text{right}} \leftarrow$ collect($\neg C \sqcap \neg D, T_{\text{right}}$)
        return $(\mathbf{CS}_{\text{left}} \cup \mathbf{CS}_{\text{right}})$
end
```

Given a TCT T, function deriveCandidateAxioms traverses the tree to collect the concept descriptions that are used as parents of the leaf-nodes. In this phase, it generates a set of concept descriptions **CS**, collecting the concepts installed in leaf-nodes (see collect). Then, it considers all pairs of elements in **CS** that are not equivalent and adds a disjointness axiom $D \sqsubseteq \neg C$, if it does not already occur in it.

The set of concept descriptions **CS** is obtained traversing the tree. The procedure collect tries to find the concept descriptions for which disjointness axioms may hold by exploring the paths from the root the leaves by collecting the concepts description installed in the internal nodes.

Note that the hierarchical nature of the approach allows one to generalize this function, controlling the maximum depth of the tree traversal with a parameter. This would produce fewer and more general axioms than the case reported above.

5 Experiments

Two experimental evaluation sessions have been performed to assess the feasibility of discovering disjointness axioms through our approach based on TCTs[1]. We also compared our method with two related statistical methods that have been recently proposed (see Sect. 2).

In the first session of experiments, we considered Web ontologies, freely available, containing disjointness axioms and describing various domains, namely: BioPax, New Testament Names (NTN), Financial, Geoskills, Monetary, and DBpedia. In the second experimental session we also considered Mutagenesis and Vicodi (that originally lack of disjointness axioms). Their principal characteristics are summarized in Table 1. The distance measure $d_2^{\mathcal{C}}$ was employed by our method, with a context of features $\mathcal{C}$ made up of the atomic concepts in each KB.

The method has been tested on the problem of (re)discovering disjointness axioms previously removed from the KB: (a) in the first session single axioms are targeted; (b) in the second session comparative experiments on versions of the ontologies enriched with further disjointness axioms (by virtue of the SDA) have been performed.

[1] Code and test ontologies are available at: http://github.com/Giuseppe-Rizzo/TCT.

Table 1. Ontologies employed in the experimental sessions

Ontology	DL language	#Concepts	#Roles	#Individuals	#Disj. Ax.'s
BioPax	$\mathcal{ALCIF}(D)$	74	70	323	85
NTN	$\mathcal{SHIF}(D)$	47	27	676	40
Financial	$\mathcal{ALCIF}(D)$	60	16	1000	113
GeoSkills	$\mathcal{ALCHOIN}(D)$	596	23	2567	378
Monetary	$\mathcal{ALCHIF}(D)$	323	247	2466	236
DBPedia3.9	$\mathcal{ALCHI}(D)$	251	132	16606	11
Mutagenesis	$\mathcal{AL}(D)$	86	5	14145	0
Vicodi	$\mathcal{ALHI}(D)$	196	10	16942	0

For each experimental session, the targeted problem and the parameter setup are described, then the outcomes are discussed.

5.1 Re-discovery of a Target Disjoint Axiom

Settings. In this session, a copy of each ontology was created removing a *target disjointness axiom*. Each copy was employed to extract a training set: given the target axiom, say $C \sqsubseteq \neg D$, we considered only individuals that belong to C and D to induce the TCTs. Table 2 lists C and D of the removed axioms for each ontology.

The experiment was repeated picking various values for the threshold ν controlling the tree growth in Algorithm 1; we report the results for $\nu = 0.9, 0.8, 0.7$. The value for δ set to 0.6. The effectiveness of the method was evaluated in terms of the *number of cases of inconsistency that were due to the addition of discovered axioms*.

Table 2. Summary of the axioms ($C \sqsubseteq \neg D$) removed from each ontology in the first experiment

Ontology	C	D
BioPax	bioSource	xref
NTN	Man	Populace ⊔ Woman ⊔ Supernatural Being
Financial	PermanentOrder	Account ⊔ Region
GeoSkills	Educational Level	Educational Pathway
Monetary	ISO3166-Country Code	ISO31813-Market Identifier Code ⊔ ISO4217-Currency Code
DBpedia3.9	Activity	Person

Outcomes. Table 3 illustrates the results of this session. Preliminarily it is worthwhile to note that the new method was able to rediscover the target disjointness axioms for all ontologies but DBPedia3.9, as discussed in the following, and it could also determine new axioms regarding concept descriptions that were equivalent to those considered in the target axioms. This case depended on the definition of ρ and on the presence of equivalence axioms in the knowledge bases: ρ assumes that all the concept names are distinct regardless of the existence of equivalence and subsumption axioms. As a result, the operator could produce potentially redundant intensional definitions. For example, in the case of BioPax $\mathsf{bioSource}$ in the target axiom is alternatively described by $\mathsf{ExternalReferenceUtilityClass} \sqcap \exists \mathsf{TAXONREF}.\top$ (with $\mathsf{ExternalReferenceUtilityClass} \sqsupseteq \mathsf{bioSource}$ and being $\mathsf{bioSource}$ the domain of the role $\mathsf{TAXONREF}$) and xref that is equivalent to $\neg\mathsf{ExternalUtilityClass} \sqcap \mathsf{PublicationXRef} \sqcap \neg\mathsf{dataSource}$. Also, in the experiments with NTN, the proposed method suggested the disjointness between $\neg\mathsf{SupernaturalBeing} \sqcap \mathsf{Person} \sqcap \mathsf{hasSex.Male}$ ($\equiv \mathsf{Man}$) and $\mathsf{SupernaturalBeing} \sqcap \mathsf{God}$ ($\equiv \mathsf{God}$, since $\mathsf{God} \sqsubseteq \mathsf{SupernaturalBeing}$).

Table 3. Number of inconsistencies (#inc.) and total number of discovered axioms (#ax's) in the first experimental session (with varying values of ν)

Ontology	TCT 0.9		TCT 0.8		TCT 0.7	
	#inc	#ax's	#inc	#ax's	#inc	#ax's
BioPax	2	53	2	53	3	52
NTN	10	70	9	73	10	75
Financial	0	125	0	126	0	127
GeoSkills	2	345	1	347	4	347
Monetary	0	432	0	432	0	433
DBPedia3.9	45	45	44	44	43	43

Moreover, the number of inconsistencies caused by the addition of axioms derived from the TCTs was quite small, especially when compared to the number of axioms predicted (see Table 3). Noticeably, in the cases of BioPax and NTN most of the instances were routed to two leaves, while the others were empty, yielding a large number of further new axioms. In such cases, the disjointness axioms involving concept descriptions that correspond to the empty clusters can be added to the knowledge base with no risk of making it inconsistent. In the experiments with larger ontologies, e.g. Financial, Monetary and GeoSkills, very few empty clusters were observed because of the larger training sets, increasing the quality of clustering and of the derived axioms.

By way of ρ the algorithm can exploit all the concepts (and roles) defined in the signature of $\mathcal{K}$, but it generally considers only a subset of individuals that occur in the ABox, $\mathbf{I} \subseteq \mathsf{Ind}(\mathcal{A})$. Therefore the heuristic wrongly favored refinements that tended to generate poor splits, sorting most of the individuals along one branch. This represents a sort of *small disjunct problem* that typically affects concept learning (with TDTs). However, we noted that this issue seldom occurred with larger training sets. Decreasing the threshold ν (that controls the depth of the branches), no significant difference was observed, because of the homogeneity of the individuals routed to the nodes.

The case of DBPedia3.9 deserves a deeper analysis. We observed that instances of a concept like Person hardly discernible in terms of the distance measure (given the choice of $\mathcal{C}$). However, as an inner concept Person was further refined into a number of sibling specializations leading to disjointness axioms involving more specific concepts. Thus, the method ended up finding disjointness axioms between pairs of more specific concepts like Activity and $\mathsf{Person} \sqcap \exists \mathsf{nationality.United\_states}$ owing to the presence in the training set of various individuals that describe American citizens. Also, the new method allowed to elicit a disjointness axiom between Activity and a concept describing non-American artists. This is a potential drawback of data-driven methods that consider general axioms including complex concepts instead of mere concept names. In our case this issue can be avoided with a more careful tuning of the parameters (thresholds) that control the growth of the tree, to prevent the involvement of overly specific concepts.

The time required for learning TCTs was quite limited spanning from few seconds (on BioPax) up to one hour (on Monetary), especially depending on the number of concepts and axioms in the ontologies. Further factors that affected the efficiency of the proposed approach were the inference services required by ρ (e.g. checking the satisfiability of computed specializations) and the computation of the medoids.

5.2 Comparison to Other Approaches Under SDA

Settings. In the second experiment, we considered two further knowledge bases (excluded from the previous experiments due to the mentioned lack of disjointness axioms): Mutagenesis and Vicodi. We considered extended versions of the ontologies reported in Table 1. Specifically, in order to test our method in comparison with two described in Sect. 2, in a scenario where the ontologies feature non trivial numbers of disjointness axioms, new versions were produced by adding disjointness axioms that involve sibling concepts in the hierarchy, according to the SDA, provided that they would keep the ontology consistent. Then, for each ontology, a fraction f of disjointness axioms was randomly removed. To determine unbiased estimates of the performance indices (i.e. independent of the specific selection of axioms removed), the empirical evaluation procedure was repeated 10 times per ontology also varying f: 20%, 50%, 70%.

Adopting the same parameter setup of the previous session, our method was compared against two related methods (see Sect. 2), respectively, one based on *Pearson's correlation coefficient* (PCC) and another exploiting *negative association rules* (NAR). As for the latter, rules were mined using Apriori; the required parameters, *minimum support rate*, *minimum confidence rate* and *maximum rule length* were set, respectively, to 10%, 50% and 3 (also in consideration of the sparseness of the instance distributions w.r.t. the concepts in the considered ontologies). The effectiveness of the methods was evaluated in terms of *the average number of inconsistencies caused by the addition of discovered axioms* (the less the better) and of *the average number of axioms that were discovered* and *rate of removed axioms re-discovered* (the larger the better).

Outcomes. In general, the method based on TCTs produced good clusterings of the training sets: the clusters were well-described by the concept descriptions in the TCTs.

As expected, the number of discovered axioms generally decreased with larger fractions of axioms removed since the resulting trees showed generally a less complex structure. Moreover the experiments showed that a nonnegligible impact on the effectiveness can come from properly tuning the threshold ν. Also, a sort of *horizon effect* was observed: the heuristic based on distance measure acted as a sort of *pre-pruning criterion* that stopped the growth of the tree too prematurely. In addition, in some case, TCTs with (nearly) empty clusters were produced. This was due to the mentioned cases of imbalanced instance distributions w.r.t. the various concepts: for example concepts with few instances are frequent in FINANCIAL, but such cases occur also in the other ontologies. However, this phenomenon was mitigated by the presence of an overall larger number of individuals to be clustered w.r.t. the previous experiment.

The outcomes reported in Table 4 show that, in absolute terms, more axioms were generally discovered by our method (considering all three choices for ν) compared with the two other methods. Moreover, the number of inconsistencies introduced (in case of direct addition to the knowledge bases) was quite limited in proportion to the number of axioms produced: for example, with MONETARY and VICODI less than the 3.5% in almost 20000 discovered axioms. This is interesting in the perspective of an integration in an ontology enrichment process: a larger variety of possibly redundant axioms may be proposed for validation with a very limited chance of introducing errors. On the other hand, the table shows that the compared approaches exhibited a more stable behavior with respect to the fractions of removed axioms f because they could discover axioms involving exclusively named concepts of the knowledge base signature whose instances are more likely to be available. Moreover, a weak correlation between two concepts is unlikely to depend on the presence of a disjointness axiom involving them. This led them also not to introduce further inconsistencies in the experiments. Inspecting sampled TCTs to gain a deeper insight into the outcomes, we could note that, for ontologies with a smaller number of concepts, such as BIOPAX and NTN, the refinement operator tended to introduce the same concept in more branches. As a consequence, a large number of axioms were discovered due to the replication of some sub-trees.

We also noted that TCTs for ontologies like DBPEDIA, VICODI and MONETARY presented concept descriptions installed at inner nodes that could be generally considered as disjoint but for few cases represented by specific

Table 4. Experimental comparison of the various approaches: average numbers of cases of inconsistency (#inc.) and total numbers of discovered axioms (#ax's)

Ontology	*f*	TCT 0.9		TCT 0.8		TCT 0.7		PCC		NAR	
		#inc	#ax's	#inc	#ax's	#inc	#ax's	#inc	#ax's	#inc	#ax's
BioPax	20%	235	3859	357	4235	365	4256	257	280	352	2990
	50%	125	3576	357	4176	432	4115				
	70%	125	3432	235	3875	417	4154				
NTN	20%	312	3128	343	3126	354	3124	32	957	376	3766
	50%	234	3023	234	3034	235	3034				
	70%	156	2987	176	2679	123	2675				
Financial	20%	76	165	87	325	96	276	124	1112	542	5366
	50%	37	143	56	307	53	259				
	70%	33	143	43	276	40	221				
GeoSkills	20%	234	14289	357	14297	432	14345	456	13384	456	13299
	50%	231	14123	356	14154	417	14256				
	70%	234	14122	358	14154	377	14187				
Monetary	20%	535	13456	573	13453	623	13460	543	13384	423	13456
	50%	315	13236	432	13236	532	13236				
	70%	247	13127	231	13127	312	13127				
Mutagenesis	20%	34	14753	43	14847	43	14978	20	2264	45	14832
	50%	23	14753	31	14753	32	14978				
	70%	23	14753	32	14753	32	14978				
Vicodi	20%	431	18231	485	18432	502	18432	475	15518	472	18721
	50%	142	18231	345	18432	467	18431				
	70%	141	18231	345	18432	312	18432				
DBPedia3.9	20%	1345	29730	1432	30143	1432	30567	1243	30470	1243	30365
	50%	1346	29730	1431	30143	1433	30567				
	70%	1343	19730	1432	30143	1432	30567				

individuals. For example, this situation applied to the concepts Actor and President in DBPedia, sharing the instance RONALD_REAGAN. The resulting axioms cannot be considered as wrong; they are intended to be submitted to the validation of a domain expert.

Considering the performance in terms of rate of rediscovered axioms (a sort of *recall* index), Table 5 shows very high rates of recovery using the TCT-based method. Even more so, it allows to express more general disjointness axioms than those obtained through the other algorithms: these tackle only the disjointness between concept names whereas from the TCTs axioms involving arbitrarily complex concept descriptions can be derived as a product of the refinement operator adopted in learning procedure. This explains why, in most cases, the number of discovered axioms, but also the number of inconsistencies, was larger with respect to those observed with the compared methods. As reported in Table 5, on average an amount of axioms could not be rediscovered by the

Table 5. Average rates (and standard deviations) of removed axioms re-discovered using the various approaches (with standard and pre-pruned TCTs)

Ontology	*f*	TCT – *standard mode*			TCT – *early stopping*		
		TCT 0.9	TCT 0.8	TCT 0.7	TCT 0.9	TCT 0.8	TCT 0.7
BioPax	20%	0.90 ± 0.12	0.76 ± 0.13	0.74 ± 0.13	0.80 ± 0.23	0.65 ± 0.23	0.70 ± 0.13
	50%	0.85 ± 0.13	0.74 ± 0.13	0.74 ± 0.13	0.63 ± 0.23	0.63 ± 0.23	0.63 ± 0.23
	70%	0.85 ± 0.13	0.74 ± 0.12	0.74 ± 0.14	0.69 ± 0.13	0.67 ± 0.13	0.66 ± 0.14
NTN	20%	0.99 ± 0.08	0.95 ± 0.06	0.95 ± 0.08	0.70 ± 0.15	0.67 ± 0.15	0.67 ± 0.14
	50%	0.97 ± 0.03	0.93 ± 0.10	0.93 ± 0.01	0.55 ± 0.13	0.54 ± 0.13	0.54 ± 0.15
	70%	0.90 ± 0.10	0.89 ± 0.11	0.89 ± 0.10	0.55 ± 0.13	0.55 ± 0.13	0.55 ± 0.13
Financial	20%	0.99 ± 0.08	0.99 ± 0.08	0.99 ± 0.08	0.60 ± 0.10	0.59 ± 0.11	0.59 ± 0.11
	50%	0.97 ± 0.03	0.97 ± 0.03	0.97 ± 0.03	0.56 ± 0.10	0.56 ± 0.10	0.56 ± 0.10
	70%	0.95 ± 0.05	0.95 ± 0.05	0.95 ± 0.05	0.56 ± 0.10	0.56 ± 0.10	0.56 ± 0.10
GeoSkills	20%	0.99 ± 0.08	0.99 ± 0.08	0.99 ± 0.08	0.70 ± 0.15	0.69 ± 0.11	0.69 ± 0.11
	50%	0.92 ± 0.10	1.00 ± 0.00	1.00 ± 0.00	0.65 ± 0.23	0.65 ± 0.23	0.65 ± 0.23
	70%	0.92 ± 0.10	0.92 ± 0.10	0.92 ± 0.10	0.65 ± 0.23	0.63 ± 0.22	0.62 ± 0.23
Monetary	20%	0.99 ± 0.08	1.00 ± 0.00	1.00 ± 0.00	0.65 ± 0.23	0.63 ± 0.20	0.62 ± 0.23
	50%	0.94 ± 0.13	1.00 ± 0.00	1.00 ± 0.00	0.63 ± 0.12	0.66 ± 0.15	0.65 ± 0.11
	70%	0.94 ± 0.13	0.91 ± 0.14	0.91 ± 0.13	0.62 ± 0.12	0.60 ± 0.13	0.60 ± 0.12
Mutagenesis	20%	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.56 ± 0.13	0.53 ± 0.14	0.51 ± 0.12
	50%	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.52 ± 0.14	0.51 ± 0.15	0.50 ± 0.11
	70%	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.50 ± 0.15	0.50 ± 0.16	0.50 ± 0.15
Vicodi	20%	0.95 ± 0.02	0.90 ± 0.08	0.90 ± 0.08	0.65 ± 0.03	0.63 ± 0.01	0.62 ± 0.01
	50%	0.95 ± 0.02	0.90 ± 0.08	0.90 ± 0.08	0.62 ± 0.03	0.62 ± 0.04	0.62 ± 0.03
	70%	0.95 ± 0.02	0.90 ± 0.08	0.90 ± 0.08	0.58 ± 0.04	0.57 ± 0.05	0.58 ± 0.04
DBPedia3.9	20%	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.70 ± 0.12	0.68 ± 0.13	0.67 ± 0.12
	50%	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.65 ± 0.23	0.68 ± 0.13	0.64 ± 0.12
	70%	0.96 ± 0.08	0.90 ± 0.08	0.90 ± 0.08	0.65 ± 0.22	0.68 ± 0.13	0.64 ± 0.12

new method. The TCT-based approach assumes the availability of the instances belonging the concepts involved in a target axiom. But in the evaluation, especially with Financial and Vicodi, some concepts were endowed with a very small number of instances (less than 10). As a result, the proposed approach could not detect cases of disjointness due to the lack of good cluster medoids. For example, in the experiments with Financial, our method was unable to discover the disjointness of the concepts Loan and Sex. For this ontology, Sex was used to model the customer's gender but no specific assertion was available: the gender was instead designed as a subconcept (Male/Female). A similar case was observed also in the experiments on Vicodi: the disjointness of Actor and Artefact could not be discovered.

Regarding the ability of discovering axioms also preventing the cases of inconsistency due to the process, one may argue that growing taller trees, thus involving very specific concept descriptions like those produced in these experiments, may turn out to be time-consuming and error-prone. To test this aspect, we checked the quality of the axioms obtained by prematurely stopping the growth of TCTs at a certain level instead of reaching the leaves namely, it was stopped if the induced axioms would introduce a case of inconsistency. Hence, Table 5 reports also the results of the experiments run with this policy implemented in the method (*early stopping* columns). As expected, the proportions of detected axioms were lower than those obtained using the standard method: no more than 80% of the removed axioms were rediscovered. This means that, although the progressive specialization of the concepts included in the disjointness axioms may lead to cases of inconsistency, working in *standard mode* the method can help elicit relevant correct axioms (inconsistencies may be avoided applying a *post-pruning* strategy aimed at preventing the production of defective axioms). Besides, we noted that the aforementioned exploration procedure was often stopped too early: in most cases, the consistency of the knowledge base was preserved by adding axioms that can be found within the 10^{th} and the 15^{th} level in the TCTs. One of the benefits deriving from the proposed approach concerns the ability to overcome one of the downsides of association rules: they cannot be considered as logical rules, rather they merely denote statistical correlations between two or more features (that hold with a degree of uncertainty).

Finally, our new method showed it could be more efficient than the one based on Apriori, that was especially slow in the step of generating the frequent patterns. The running-time for inducing TCTs spanned from less than a minute to hours for the various ontologies, while the time required by the association rule mining was larger in most cases. Note that this also depends on the maximum length of the rules: mining longer rules (to discover more axioms) would make the method infeasible.

6 Conclusions and Outlook

In this work, we have illustrated the terminological cluster trees, an extension of terminological decision trees [8] that, unlike these supervised classification models, aims at solving an unsupervised problem: clustering individual resources occurring in Web ontologies. As an application, we have cast the task of discovering disjointness axioms as the mentioned clustering problem and have proposed a solution that exploits the new models. In the presented empirical evaluation the effectiveness of the proposed approach was tested. Compared to related unsupervised approaches, the new method proved to be able to discover disjointness axioms involving complex concept descriptions exploiting the ontology as a source of background knowledge, unlike the other methods based on the statistical correlation between instances.

This work can be extended along various directions. One regards the investigation of other distances measures for individuals and different notions of separation between clusters. In addition, this approach can be integrated with other machine learning-based frameworks for ontology engineering, such as DL-Learner [14], as a service for enriching the terminology of lightweight ontologies.

The method can be further improved by introducing a post-pruning step for better tackling the problem of empty-clusters. Finally, the empirical evaluation may be extended considering further methods and well-modeled ontologies endowed with disjointness axioms which seem hard to find.

References

1. Aggarwal, C.C., Reddy, C.K.: Data Clustering: Algorithms and Applications, 1st edn. Chapman & Hall/CRC, Boca Raton (2013)
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook, 2nd edn. Cambridge University Press, Cambridge (2007)
3. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing description logic knowledge bases using formal concept analysis. In: Veloso, M. (ed.) Proceedings of IJCAI 2007, pp. 230–235. AAAI Press, Menlo Park (2007)
4. Blockeel, H., De Raedt, L.: Top-down induction of first-order logical decision trees. Artif. Intell. **101**(1–2), 285–297 (1998)
5. Cornet, R., Abu-Hanna, A.: Usability of expressive description logics - a case study in UMLS. In: Kohane, I. (ed.) Proceedings of AMIA 2002, pp. 180–184. AMIA (2002)
6. d'Amato, C., Fanizzi, N., Esposito, F.: Query answering and ontology population: an inductive approach. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 288–302. Springer, Heidelberg (2008). doi:10.1007/978-3-540-68234-9_23
7. De Raedt, L., Blockeel, H.: Using logical decision trees for clustering. In: Lavrač, N., Džeroski, S. (eds.) ILP 1997. LNCS, vol. 1297, pp. 133–140. Springer, Heidelberg (1997). doi:10.1007/3540635149_41
8. Fanizzi, N., d'Amato, C., Esposito, F.: Induction of concepts in web ontologies through terminological decision trees. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6321, pp. 442–457. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15880-3_34
9. Fanizzi, N., d'Amato, C.: A hierarchical clustering method for semantic knowledge bases. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007. LNCS (LNAI), vol. 4694, pp. 653–660. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74829-8_80
10. Fanizzi, N., d'Amato, C., Esposito, F.: Evolutionary conceptual clustering based on induced pseudo-metrics. Int. J. Semant. Web Inf. Syst. **4**(3), 44–67 (2008)
11. Fleischhacker, D., Völker, J.: Inductive learning of disjointness axioms. In: Meersman, R., et al. (eds.) OTM 2011. LNCS, vol. 7045, pp. 680–697. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25106-1_20
12. Haase, P., Völker, J.: Ontology learning and reasoning: dealing with uncertainty and inconsistency. In: da Costa, P., et al. (eds.) Uncertainty Reasoning for the Semantic Web I. LNCS, vol. 5327, pp. 366–384. Springer, Heidelberg (2008)
13. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, San Rafael (2011)
14. Hellmann, S., Lehmann, J., Auer, S.: Learning of OWL class descriptions on very large knowledge bases. Int. J. Semant. Web Inf. **5**(2), 25–48 (2009)
15. Lehmann, J., Bühmann, L.: ORE - a tool for repairing and enriching knowledge bases. In: Patel-Schneider, P.F., et al. (eds.) ISWC 2010. LNCS, vol. 6497, pp. 177–193. Springer, Heidelberg (2010). doi:10.1007/978-3-642-17749-1_12

16. Rizzo, G., d'Amato, C., Fanizzi, N., Esposito, F.: Induction of terminological cluster trees. In: Bobillo, F., et al. (ed.) Proceedings of URSW 2016. CEUR Workshop Proceedings, vol. 1665, pp. 49–60. CEUR-WS.org (2016)
17. Schlobach, S.: Debugging and semantic clarification by pinpointing. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 226–240. Springer, Heidelberg (2005). doi:10.1007/11431053_16
18. Völker, J., Fleischhacker, D., Stuckenschmidt, H.: Automatic acquisition of class disjointness. J. Web Semant. **35**(P2), 124–139 (2015)
19. Völker, J., Niepert, M.: Statistical schema induction. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., Leenheer, P., Pan, J. (eds.) ESWC 2011. LNCS, vol. 6643, pp. 124–138. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21034-1_9
20. Völker, J., Vrandečić, D., Sure, Y., Hotho, A.: Learning disjointness. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 175–189. Springer, Heidelberg (2007). doi:10.1007/978-3-540-72667-8_14
21. Wang, T.D., Parsia, B., Hendler, J.: A survey of the web ontology landscape. In: Cruz, I., et al. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 682–694. Springer, Heidelberg (2006). doi:10.1007/11926078_49