A Hierarchical Clustering Procedure for Semantically Annotated Resources

Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito

Dipartimento di Informatica – Università degli Studi di Bari
Campus Universitario, Via Orabona 4, 70125 Bari, Italy
{fanizzi|claudia.damato|esposito}@di.uniba.it

Abstract. A clustering method is presented which can be applied to relational knowledge bases. It can be used to discover interesting groupings of resources through their (semantic) annotations expressed in the standard languages employed for modeling concepts in the Semantic Web. The method exploits a simple (yet effective and language-independent) semi-distance measure for individuals, that is based on the resource semantics w.r.t. a number of dimensions corresponding to a committee of features represented by a group of concept descriptions (discriminating features). The algorithm is an fusion of the classic Bisecting k-Means with approaches based on medoids since they are intended to be applied to relational representations. We discuss its complexity and the potential applications to a variety of important tasks.

1 Learning Methods for Concept Languages

In the inherently distributed applications related to the Semantic Web (henceforth SW) there is an extreme need of automatizing those activities which are more burdensome for the knowledge engineer, such as ontology construction, matching and evolution. Such an automatization may be assisted by crafting supervised or unsupervised methods for the specific representations of the SW field (RDF through OWL).

In this work, we investigate on unsupervised learning for knowledge bases expressed in such standard concept languages. In particular, we focus on the problem of conceptual clustering of semantically annotated resources. The benefits of *conceptual clustering* [17] in the context of semantically annotated knowledge bases are manifold:

- *concept formation*: clustering annotated resources enables the definition of new emerging concepts on the grounds of the primitive concepts asserted in a knowledge base;
- *evolution*: supervised methods can exploit these clusters to induce new concept definitions or to refining existing ones;
- *discovery and ranking*: intensionally defined groupings may speed-up the task of search and discovery; a hierarchical clustering also suggests criteria for ranking the retrieved resources.

Essentially, many existing clustering methods are based on the application of similarity (or density) measures defined over a fixed set of attributes of the domain objects. Classes of objects are taken as collections that exhibit low interclass similarity (density) and high intraclass similarity (density). Often these methods cannot into account any form of *background knowledge* that could characterize object configurations by means of global concepts and semantic relationship. This hinders the interpretation of the outcomes of these methods which is crucial in the SW perspective which foresees sharing and reusing the produced knowledge in order to enable forms of semantic interoperability.

Thus, early conceptual clustering methods aimed at defining groups of objects through conjunctive descriptions based on selected attributes [17]. In the perspective, the expressiveness of the language adopted for describing objects and clusters (concepts) is equally important. Alternative approaches, particularly suitable to concept languages and terminological representations, have pursued a different way for attacking the problem, devising logic-based methods [12, 7]. Yet it has been pointed out that these methods may suffer from noise in the data.

This motivates our investigation on similarity-based clustering methods which can be more noise-tolerant, still saving the advantages of conceptual clustering. Specifically we propose a multi-relational extension of effective clustering techniques, which is tailored for the SW context. It is intended for grouping similar resources w.r.t. a semantic dissimilarity measure which allows for discovering new concepts. Specifically, our relational method derives from the *Bisecting k-means* algorithm [10], a well-known partitional clustering method.

From a technical viewpoint, upgrading existing algorithms to work on multi-relational representations, like the concept languages used in the SW, required novel similarity measures that are suitable for such representations. In particular, as for the original method, one may fix a given number k of clusters of interest, yet this may be hard when scarce knowledge about the domain is available. As an alternative, a partitional method may be employed up to reaching a minimal threshold value for cluster *quality* (many measures have been proposed in the literature [9]) which makes any further bisections useless.

In this setting, the notion of means that characterizes the algorithms descending from *k-means* and *EM* [10] developed for numeric (or just ordinal) features. In our case we recur to the notion of *medoids* (like in algorithm *PAM* [11]) as central individuals in a cluster. Another theoretical problem is posed by the *Open World Assumption* (OWA) that is generally made in the target context, differently from the *Closed World Assumption* (CWA) which is often made when performing machine learning or query-answering tasks.

The notion of similarity to be employed has to deal with the rich representations of semantically annotated resources. Therefore we developed a measure which could be used specifically for the SW standard representations (see below). As pointed out in a seminal paper [3] on similarity measures for DLs, most of the existing measures focus on the similarity of atomic concepts within hierarchies or simple ontologies. Moreover, they have been conceived for assessing *concept*

similarity, whereas, for our purposes, a notion of similarity between *individuals* is required.

Recently, dissimilarity measures for specific DLs have been proposed [4]. Although they turned out to be quite effective for the inductive tasks, they are still partly based on structural criteria which determine their main weakness: they are hardly scalable to deal with standard languages used in the current knowledge management frameworks. Therefore, we have devised a family of dissimilarity measures for semantically annotated resources, which can overcome the aforementioned limitations. Following some ideas introduced in [16], we present a new family of measures that is suitable for a wide range of ontology languages since it is merely based on the discernibility of the input individuals with respect to a fixed set of features represented by concept definitions (features). As such the new measures are not absolute, yet they depend on the knowledge base they are applied to.

The remainder of the paper is organized as follows. Sect. 2 presents the basics representation and the novel semantic similarity measure adopted with the clustering algorithm. This algorithm is presented and discussed in Sect. 3. After Sect. 4 concerning the related work, possible developments are finally examined in Sect. 5.

2 Semantic Distance Measures

2.1 Preliminaries on the Reference Representation

One of the strong points of our method is that it does not rely on a particular language for semantic annotations. Hence, in the following, we assume that resources, concepts and their relationship may be defined in terms of a generic ontology language that may be mapped to some DL language with the standard model-theoretic semantics (see the handbook [1] for a thorough reference).

In this context, a *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ contains a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. $\mathcal{T}$ is a set of concept definitions. $\mathcal{A}$ contains assertions (facts, data) concerning the world state. Moreover, normally the *unique names assumption* is made on the ABox individuals[1] therein. The set of the individuals occurring in $\mathcal{A}$ will be denoted with $\mathsf{Ind}(\mathcal{A})$.

As regards the inference services, like all other instance-based methods, our procedure may require performing *instance-checking*, which amounts to determining whether an individual, say a, belongs to a concept extension, i.e. whether $C(a)$ holds for a certain concept C.

2.2 A Semantic Semi-Distance for Individuals

Moreover, for our purposes, we need a function for measuring the similarity of individuals rather than concepts. It can be observed that individuals do not have a syntactic structure that can be compared. This has led to lifting them to the

[1] Individuals can be assumed to be identified by their own URI.

concept description level before comparing them (recurring to the approximation of the *most specific concept* of an individual w.r.t. the ABox).

For the clustering procedure specified in Sect. 3, we have developed a new measure with a definition that totally depends on semantic aspects of the individuals in the knowledge base.

On a semantic level, similar individuals should behave similarly with respect to the same concepts. We introduce a novel measure for assessing the similarity of individuals in a knowledge base, which is based on the idea of comparing their semantics along a number of dimensions represented by a committee of concept descriptions. Following the ideas borrowed from ILP [16] and *multi-dimensional scaling*, we propose the definition of totally semantic distance measures for individuals in the context of a knowledge base.

The rationale of the new measure is to compare them on the grounds of their behavior w.r.t. a given set of hypotheses, that is a collection of concept descriptions, say $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, which stands as a group of discriminating *features* expressed in the language taken into account.

In its simple formulation, a family of distance functions for individuals inspired to Minkowski's distances can be defined as follows:

Definition 2.1 (family of measures). *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base. Given set of concept descriptions $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, a family of functions*

$$d_p^{\mathsf{F}} : \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mapsto \mathbb{R}$$

defined as follows:

$$\forall a, b \in \mathsf{Ind}(\mathcal{A}) \quad d_p^{\mathsf{F}}(a, b) := \frac{1}{m} \left[\sum_{i=1}^{m} \mid \pi_i(a) - \pi_i(b) \mid^p \right]^{1/p}$$

where $p > 0$ and $\forall i \in \{1, \ldots, m\}$ the projection function *π_i is defined by:*

$$\forall a \in \mathsf{Ind}(\mathcal{A}) \quad \pi_i(a) = \begin{cases} 1 & F_i(x) \in \mathcal{A} \\ 0 & \neg F_i(x) \in \mathcal{A} \\ 1/2 & \textit{otherwise} \end{cases}$$

The superscript F will be omitted when the set of hypotheses is fixed.

As an alternative, the definition of the measures can be made more accurate by considering entailment rather than the simple ABox look-up, when determining the values of the projection functions:

$$\forall a \in \mathsf{Ind}(\mathcal{A}) \quad \pi_i(a) = \begin{cases} 1 & \mathcal{K} \models F_i(x) \\ 0 & \mathcal{K} \models \neg F_i(x) \\ 1/2 & \textit{otherwise} \end{cases}$$

In particular, for the sake of saving computational resources, we have considered the following measures in the experiments: $\forall a, b \in \mathsf{Ind}(\mathcal{A})$

$$d_1(a, b) := \frac{1}{m} \sum_{i=1}^{m} \mid \pi_i(a) - \pi_i(b) \mid \quad \text{or} \quad d_2(a, b) := \frac{1}{m} \sqrt{\sum_{i=1}^{m} (\pi_i(a) - \pi_i(b))^2}$$

In order to clarify the application of the presented measures the following example is considered.

Example 2.1. Let us consider the knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ reported below.
Primitive Concepts: $N_C = \{\mathsf{Female},\ \mathsf{Male},\ \mathsf{Human}\}$.
Primitive Roles: $N_R = \{\mathsf{HasChild},\ \mathsf{HasParent},\ \mathsf{HasGrandParent},\ \mathsf{HasUncle}\}$.

T-Box: $\mathcal{T} = \{$ $\mathsf{Woman} \equiv \mathsf{Human} \sqcap \mathsf{Female}$, $\mathsf{Man} \equiv \mathsf{Human} \sqcap \mathsf{Male}$
$\mathsf{Parent} \equiv \mathsf{Human} \sqcap \exists \mathsf{HasChild}.\mathsf{Human}$, $\mathsf{Mother} \equiv \mathsf{Woman} \sqcap \mathsf{Parent}$ $\exists \mathsf{HasChild}.\mathsf{Human}$
$\mathsf{Father} \equiv \mathsf{Man} \sqcap \mathsf{Parent}$, $\mathsf{Child} \equiv \mathsf{Human} \sqcap \exists \mathsf{HasParent}.\mathsf{Parent}$,
$\mathsf{Grandparent} \equiv \mathsf{Parent} \sqcap \exists \mathsf{HasChild}.(\ \exists\ \mathsf{HasChild}.\mathsf{Human})$,
$\mathsf{Sibling} \equiv \mathsf{Child} \sqcap \exists \mathsf{HasParent}.(\ \exists\ \mathsf{HasChild} \geq 2)$,
$\mathsf{Niece} \equiv \mathsf{Human} \sqcap \exists \mathsf{HasGrandParent}.\mathsf{Parent} \sqcup \exists \mathsf{HasUncle}.\mathsf{Uncle}$,
$\mathsf{Cousin} \equiv \mathsf{Niece} \sqcap \exists \mathsf{HasUncle}.(\exists\ \mathsf{HasChild}.\mathsf{Human})$ $\}$.

A-Box: $\mathcal{A} = \{$Woman(Claudia), Woman(Tiziana), Father(Leonardo), Father(Antonio), Father(AntonioB), Mother(Maria), Mother(Giovanna), Child(Valentina), Sibling(Martina), Sibling(Vito), Mother(Tiziana), HasParent(Claudia,Giovanna), HasParent(Leonardo,AntonioB), HasParent(Martina,Maria), HasParent(Giovanna,Antonio), HasParent(Vito,AntonioB), HasParent(Tiziana,Giovanna), HasParent(Tiziana,Leonardo), HasParent(Valentina,Maria), HasParent(Maria,Antonio), HasSibling(Leonardo,Vito), HasSibling(Martina,Valentina), HasSibling(Giovanna,Maria), HasSibling(Vito,Leonardo), HasSibling(Tiziana,Claudia), HasSibling(Valentina,Martina), HasChild(Leonardo,Tiziana), HasChild(Antonio,Giovanna), HasChild(Antonio,Maria), HasChild(Giovanna,Tiziana), HasChild(Giovanna,Claudia), HasChild(AntonioB,Vito), HasChild(AntonioB,Leonardo), HasChild(Maria,Valentina), HasUncle(Martina,Giovanna), HasUncle(Valentina,Giovanna) $\}$

Considered this knowledge base and a feature set $\mathsf{F} = \{\mathsf{Woman}, \mathsf{Man}, \mathsf{Parent}, \mathsf{Sibling}, \mathsf{Child}\}$ it is possible to compute the similarity value between the individuals: Claudia and Tiziana as:

$$d_1(\mathsf{Claudia}, \mathsf{Tiziana}) := \frac{1}{5} \sum_{i=1}^{5} \mid \pi_i(\mathsf{Claudia}) - \pi_i(\mathsf{Tiziana}) \mid =$$
$$= \frac{1}{5} \cdot (\mid 1-1 \mid + \mid 0-0 \mid + \mid 0-1 \mid + \mid 1-1 \mid + \mid 1-1 \mid) = \frac{1}{5} = 0.2$$

2.3 Discussion

It is easy to prove that these functions have the standard properties for semi-distances:

Proposition 2.1 (semi-distance). *For a fixed feature set and $p > 0$, given any three instances $a, b, c \in \mathsf{Ind}(\mathcal{A})$. it holds that:*

1. $d_p(a, b) > 0$
2. $d_p(a, b) = d_p(b, a)$

3. $d_p(a,c) \leq d_p(a,b) + d_p(b,c)$

Proof.

1. *trivial*
2. *trivial*
3. *Noted that*

$$
\begin{aligned}
(d_p(a,c))^p &= \frac{1}{m^p}\sum_{i=1}^{m} \mid \pi_i(a) - \pi_i(c) \mid^p \\
&= \frac{1}{m^p}\sum_{i=1}^{m} \mid \pi_i(a) - \pi_i(b) + \pi_i(b) - \pi_i(c) \mid^p \\
&\leq \frac{1}{m^p}\sum_{i=1}^{m} \mid \pi_i(a) - \pi_i(b) \mid^p + \frac{1}{m^p}\sum_{i=1}^{m} \mid \pi_i(b) - \pi_i(c) \mid^p \\
&\leq (d_p(a,b))^p + (d_p(b,c))^p \leq (d_p(a,b) + d_p(b,c))^p
\end{aligned}
$$

then the property follows for the monotonicity of the power function.

It cannot be proved that $d_p(a,b) = 0$ iff $a = b$. This is the case of *indiscernible* individuals with respect to the given set of hypotheses F.

Compared to other proposed distance (or dissimilarity) measures [3], the presented function does not depend on the constructors of a specific language, rather it requires only retrieval or instance-checking service used for deciding whether an individual is asserted in the knowledge base to belong to a concept extension (or, alternatively, if this could be derived as a logical consequence).

Note that the π_i functions ($\forall i = 1, \ldots, m$) for the training instances, that contribute to determine the measure with respect to new ones, can be computed in advance thus determining a speed-up in the actual computation of the measure. This is very important for the measure integration in algorithms which massively use this distance, such as all instance-based methods.

The underlying idea for the measure is that similar individuals should exhibit the same behavior w.r.t. the concepts in F. Here, we make the assumption that the feature-set F represents a sufficient number of (possibly redundant) features that are able to discriminate really different individuals. Experimentally, we could obtain good results by using the very set of both primitive and defined concepts found in the ontology. The choice of the concepts to be included – *feature selection* – may be crucial. We have devised a specific optimization algorithms founded in *genetic programming* and *simulated annealing* (whose presentation goes beyond the scope of this work) which are able to find optimal choices of discriminating concept committees.

3 Grouping Individuals by Hierarchical Clustering

The conceptual clustering procedure implemented in our method works top-down, starting with one universal cluster grouping all instances. Then it iteratively finds two clusters bisecting an existing one up to the desired number of

clusters is reached. This algorithm can be thought as producing a dendrogram levelwise: the number of levels coincides with the number of clusters. It can be very fast.

3.1 The Algorithm

In particular our algorithm can be ascribed to the category of the heuristic partitioning algorithms such as K-MEANS and EM [9, 10]. Each cluster is represented by the center of the cluster. In our setting we will consider the notion of medoid as a notion of cluster center since our distance measure works on a categorical feature-space. In particular it can be seen as a hierarchical extension of the PAM algorithm (*Partition Around Medoids* [11]): each cluster is represented by one of the individuals in the cluster, the medoid, i.e., in our case, the one with the lowest average distance w.r.t. all the others individuals in the cluster. The bi-partition is repeated level-wise producing a dendrogram.

Fig. 1 reports a sketch of our algorithm. It essentially consists of two nested loops: the outer one computes a new level of the resulting dendrogram and it is repeated until the desired number of clusters is obtained (which corresponds to the latest level; the inner loop consists of a run of the PAM algorithm at the current level.

Per each level, the next worst cluster is selected (*selectWorstCluster*() function) on the grounds of its quality, e.g. the one endowed with the least average inner similarity (or cohesiveness [17]). This cluster is candidate to being parted in two. The partition is constructed around two medoids initially chosen (*selectMostDissimilar*() function) as the most dissimilar elements in the cluster and then iteratively adjusted in the inner loop. In the end, the candidate cluster is replaced by the newly found parts at the next level of the dendrogram.

The inner loop basically resembles to a 2-means (or EM) algorithm, where medoids are considered instead of means which can hardly be defined in symbolic computations. Then, the classical two steps are performed in an iteration:

E step given the current medoids, the first distributes the other individuals in one of the two partitions under construction on the grounds of their similarity w.r.t. either medoid;

M step given the bipartition obtained by *distribute*(), this second step computes the new medoids for either cluster. These tend to change on each iteration until eventually they converge to a stable couple (or when a maximum number of iteration have been performed).

The medoid of a group of individuals is the individual that has the lowest distance w.r.t. the others. Formally. given a cluster $C = \{a_1, a_2, \ldots, a_n\}$, the medoid is defined:

$$m = \text{medoid}(C) = \operatorname*{argmin}_{a \in C} \sum_{j=1}^{n} d(a, a_j)$$

```
input  allIndividuals: set of individuals
       k: number of clusters;
       maxIterations: max number of inner iterations;
output clusterVector: array [1..k] of sets of clusters

level := 0;
clusterVector[1] := allIndividuals;
repeat
       ++level;
       cluster2split := selectWorstCluster(clusterVector[level]);
       iterCount := 0;
       stableConfiguration := false;
       (newMedoid1,newMedoid2) := selectMostDissimilar(cluster2split);
       repeat
            ++iterCount;
            // E step
            (medoid1,medoid2) := (newMedoid1,newMedoid2);
            (cluster1,cluster2) := distribute(cluster2split,medoid1,medoid2);
            // M step
            newMedoid1 := medoid(cluster1);
            newMedoid2 := medoid(cluster2);
            stableConfiguration := (medoid1 = newMedoid1) ∧ (medoid2 = newMedoid2);
       until stableConfiguration ∨ (iterCount = maxIterations);
       clusterVector[level+1] := replace(cluster2split,cluster1,cluster2,clusterVector[level]);
until (level = k);
```

Fig. 1. The Hierarchical Bisecting around Medoids Algorithm.

Each node of the tree (a cluster) may be labeled with an intensional concept definition which characterizes the individuals in the given cluster while discriminating those in the twin cluster at the same level. Labeling the tree-nodes with concepts can be regarded as a number of supervised learning problems in the specific multi-relational representation targeted in our setting. As such it deserves specific solutions that are suitable for the DL languages employed.

A straightforward solution may be found, for DLs that allow for the computation of (an approximation of) the *most specific concept* (msc) and *least common subsumer* (lcs) [2] (such as $\mathcal{ALC}$). This may involve the following steps:
given a cluster of individuals node_j

- **for each** individual $a_i \in \mathsf{node}_j$ **do**
  compute $M_i := \mathsf{msc}(a_i)$ w.r.t. $\mathcal{A}$;
- **let** $\mathsf{MSCs}_j := \{M_i \mid a_i \in \mathsf{node}_j\}$;
- **return** $\mathsf{lcs}(\mathsf{MSCs}_j)$

As an alternative, algorithms for learning concept descriptions expressed in DLs may be employed [5]. Indeed, concept formation can be cast as a supervised learning problem: once the two clusters at a certain level have been found, where the members of a cluster are considered as positive examples and the members

of the dual cluster as negative ones. Then any concept learning method which can deal with this representation may be utilized for this new task.

3.2 Discussion

The representation of centers by means of medoids has two advantages. First, it presents no limitations on attributes types, and, second, the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster and, therefore, it is lesser sensitive to the presence of outliers. In K-MEANS case a cluster is represented by its centroid, which is a mean (usually weighted average) of points within a cluster. This works conveniently only with numerical attributes and can be negatively affected by a single outlier.

A PAM algorithm has several favorable properties. Since it performs clustering with respect to any specified metric, it allows a flexible definition of similarity. This flexibility is particularly important in biological applications where researchers may be interested, for example, in grouping correlated or possibly also anti-correlated elements. Many clustering algorithms do not allow for a flexible definition of similarity, but allow only Euclidean distance in current implementations. In addition to allowing a flexible distance metric, a PAM algorithm has the advantage of identifying clusters by the medoids. Medoids are robust representations of the cluster centers that are less sensitive to outliers than other cluster profiles, such as the cluster means of K-MEANS. This robustness is particularly important in the common context that many elements do not belong exactly to any cluster, which may be the case of the membership in DL knowledge bases, which may be not ascertained given the OWA.

4 Related Work

The unsupervised learning setting presented in this paper is mainly is based on two factors: the semantic similarity measure and the clustering method. In the following, we briefly discuss sources of inspiration and related approaches.

4.1 Semantic Similarity Measures

As mentioned in the first section, various attempts to define semantic similarity (or dissimilarity) measures for concept languages have been made, yet they have still a limited applicability to simple languages [3] or they are not completely semantic depending also on the structure of the descriptions [4].

Our measure is mainly based on Minkowski's measure and on a method for distance induction developed by Sebag [16] in the context of machine learning (and *inductive logic programming*). It is shown that the induced measure could be accurate when employed for classification tasks even though set of features (hypotheses) to be used were not the optimal ones (or they were redundant).

A source of inspiration was also *rough sets* theory [15] which aims at the formal definition of vague sets by means of their approximations determined

by an indiscernibility relationship. Hopefully, these methods developed in this context will help solving the open points of our framework (see the next section) and suggest new ways to treat uncertainty.

Another related metric was defined [14] for the Herbrand interpretations of logic clauses as induced from a metric on ground atoms. Specifically, it may be employed to assess the dissimilarity of individuals by deriving a related *most specific concept* description (MSC) [1] accounting for them.

4.2 Clustering Procedures

Our algorithm adapts to the specific representations devised for the SW context a combination of Bisecting K-Means clustering and the approaches based on medoids. Specifically, in k-medoids methods each cluster is represented by one of its points. Two early versions of k-medoid approach are the algorithms PAM (*Partitioning Around Medoids*) and CLARA (*Clustering LARge Applications*) [11]. PAM is an iterative optimization method that combines relocation of points between perspective clusters with re-nominating the points as potential medoids. The guiding principle for the process is the effect on an objective function, which, obviously, is a costly strategy. CLARA uses several samples of points, which are subjected to PAM. The whole dataset is assigned to resulting medoids, the objective function is computed, and the best system of medoids is retained.

CLARANS algorithm (*Clustering Large Applications based upon RANdomized Search*) [13] was introduced in the context of spatial databases. A graph is considered whose nodes are sets of k medoids and an edge connects two nodes if they differ by exactly one medoid. While CLARA compares very few neighbors corresponding to a fixed small sample, CLARANS uses random search to generate neighbors by starting with an arbitrary node and randomly checking maxneighbor neighbors. If a neighbor represents a better partition, the process continues with this new node. Otherwise a local minimum is found, and the algorithm restarts until a certain number of local minima is found. The best node (i.e. a set of medoids) is returned for the formation of a resulting partition. The complexity of CLARANS is $O(N^2)$ in terms of number of points. Ester et al. [6] extended CLARANS to very large spatial databases. Our algorithm may be considered an extension of the simpler forms of k-medoids to a hierarchical case. This allows also do determine a good estimate of the number of clusters

Further comparable clustering methods are those based on an indiscernibility relationship [8]. While in our method this idea is embedded in the semi-distance measure (and the choice of the committee of concepts), these algorithms are based on an iterative refinement of an equivalence relationship which induces clusters as equivalence classes.

5 Conclusions and Future Work

This work has presented a clustering for (multi-)relational representations which are standard in the SW field. Namely, it can be used to discover interesting groupings of semantically annotated resources in a wide range of concept languages.

The method exploits a novel dissimilarity measure, that is based on the resource semantics w.r.t. a number of dimensions corresponding to a committee of features represented by a group of concept descriptions (discriminating features). The algorithm, is an adaptation of the classic bisecting k-means to complex representations typical of the ontology in the SW. We have discussed its complexity and the potential applications to a variety of important tasks.

In order to exploit the method, the underlying application needs only to model the knowledge base according to this ontology language, thus, for instance, a SW service registry may be searched provided that services are described in the standard representations defined on top of OWL (e.g. OWL-S or WSML).

Ongoing work concerns the mentioned feature selection task. Namely, we aim at inducing an optimal set of concepts for the distance measure by means of randomized algorithms based on genetic programming and simulated annealing. Furthermore, also the clustering process itself may be carried out by means of a randomized method based on the same approaches.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
2. F. Baader and R. Küsters. Non-standard inferences in description logics: The story so far. In D. Gabbay, S. S. Goncharov, and M. Zakharyaschev, editors, *Mathematical Problems from Applied Logic. New Logics for the XXIst Century*, volume 4 of *International Mathematical Series*. Kluwer/Plenum Publishers, 2005.
3. A. Borgida, T. Walsh, and H. Hirsh. Towards measuring similarity in description logics. In I. Horrocks, U. Sattler, and F. Wolter, editors, *Working Notes of the International Description Logics Workshop*, volume 147 of *CEUR Workshop Proceedings*, Edinburgh, UK, 2005.
4. C. d'Amato, N. Fanizzi, and F. Esposito. Reasoning by analogy in description logics through instance-based learning. In G. Tummarello, P. Bouquet, and O. Signore, editors, *Proceedings of Semantic Web Applications and Perspectives, 3rd Italian Semantic Web Workshop, SWAP2006*, volume 201 of *CEUR Workshop Proceedings*, Pisa, Italy, 2006.
5. F. Esposito, N. Fanizzi, L. Iannone, I. Palmisano, and G. Semeraro. Knowledge-intensive induction of terminologies from metadata. In F. van Harmelen, S. McIlraith, and D. Plexousakis, editors, *ISWC2004, Proceedings of the 3rd International Semantic Web Conference*, volume 3298 of *LNCS*, pages 441–455. Springer, 2004.
6. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. In *Proceedings of the 2nd Conference of ACM SIGKDD*, pages 226–231, 1996.
7. N. Fanizzi, L. Iannone, I. Palmisano, and G. Semeraro. Concept formation in expressive description logics. In J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, editors, *Proceedings of the 15th European Conference on Machine Learning, ECML2004*, volume 3201 of *LNAI*, pages 99–113. Springer, 2004.
8. S. Hirano and S. Tsumoto. An indiscernibility-based clustering method. In X. Hu, Q. Liu, A. Skowron, T. Y. Lin, R. Yager, and B. Zhang, editors, *2005 IEEE International Conference on Granular Computing*, pages 468–473. IEEE, 2005.

9. A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ, 1988.
10. A. Jain, M. Murty, and P. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.
11. L. Kaufman and R. P.J. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, 1990.
12. J.-U. Kietz and K. Morik. A polynomial approach to the constructive induction of structural knowledge. *Machine Learning*, 14(2):193–218, 1994.
13. R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proceedings of the 20th Conference on Very Large Databases, VLDB94*, pages 144–155, 1994.
14. S.-H. Nienhuys-Cheng. Distances and limits on herbrand interpretations. In D. Page, editor, *Proceedings of the 8th International Workshop on Inductive Logic Programming, ILP98*, volume 1446 of *LNAI*, pages 250–260. Springer, 1998.
15. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, 1991.
16. M. Sebag. Distance induction in first order logic. In S. Džeroski and N. Lavrač, editors, *Proceedings of the 7th International Workshop on Inductive Logic Programming, ILP97*, volume 1297 of *LNAI*, pages 264–272. Springer, 1997.
17. R. E. Stepp and R. S. Michalski. Conceptual clustering of structured objects: A goal-oriented approach. *Artificial Intelligence*, 28(1):43–69, Feb. 1986.