Conceptual Clustering and Its Application to Concept Drift and Novelty Detection

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. The paper presents a clustering method which can be applied to populated ontologies for discovering interesting groupings of resources therein. The method exploits a simple, yet effective and language-independent, semi-distance measure for individuals, that is based on their underlying semantics along with a number of dimensions corresponding to a set of concept descriptions (discriminating features committee). The clustering algorithm is a partitional method and it is based on the notion of medoids w.r.t. the adopted semi-distance measure. Eventually, it produces a hierarchical organization of groups of individuals. A final experiment demonstrates the validity of the approach using absolute quality indices. We propose two possible exploitations of these clusterings: concept formation and detecting concept drift or novelty.

1 Introduction

In the perspective of automatizing the most burdensome activities for the knowledge engineer, such as ontology construction, matching and evolution, they may be assisted by supervised or unsupervised methods crafted for the standard representations adopted in the Semantic Web (SW) context and founded in *Description Logics* (DLs).

In this work, we investigate unsupervised learning for populated ontologies. Specifically, we focus on the problem of *conceptual clustering* [27] of semantically annotated resources, that amounts to grouping them into clusters according to some criteria (e.g. similarity). The benefits of conceptual clustering in the context of knowledge bases maintenance are manifold. Clustering resources enables the definition of new emerging concepts (*concept formation*) on the grounds of those already defined (intensionally or extensionally) in the knowledge base; supervised methods can then exploit these clusters to induce new concept definitions or to refining existing ones (*ontology evolution*); intensionally defined groupings may speed-up the task of search and *discovery* [6]; a hierarchical clustering also suggests criteria for *ranking* the retrieved resources.

Essentially, most of the 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). More rarely these methods are able to exploit (declarative) forms of *prior* or *background knowledge* to characterize the clusters with intensional definitions. This hinders the interpretation of the outcomes of these methods which is crucial in the SW perspective that should enforce semantic

S. Bechhofer et al.(Eds.): ESWC 2008, LNCS 5021, pp. 318–332, 2008.

interoperability through knowledge sharing and reuse. Thus, specific conceptual clustering methods have to be taken into account, such as those focussing on the definition of groups of objects through conjunctive descriptions based on selected attributes [27]. More recent related works are based on similarity measures for clausal spaces [18], yet the expressiveness of these representations is incomparable w.r.t. DLs [3]. Also the underlying semantics is different since the *Open World Assumption* (OWA) is made on DL languages, whereas the *Closed World Assumption* (CWA) is the standard in machine learning and data mining.

Regarding dissimilarity measures in DL languages, as pointed out in a seminal paper [4], 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 accomplishing inductive tasks, such as clustering, a notion of similarity between *individuals* is required. Recently, dissimilarity measures for specific DLs have been proposed [5]. Although they turned out to be quite effective for the inductive tasks, they are still partly based on structural criteria which makes them fail to fully grasp the underlying semantics and hardly scale to more complex ontology languages that are commonly adopted in the SW context.

Therefore, we have devised a family of dissimilarity measures for semantically annotated resources, which can overcome the aforementioned limitations. Namely, we adopt a new family of measures [8] that is suitable for a wide range of languages since it is merely based on the discernibility of the individuals with respect to a fixed set of features (henceforth a *committee*) represented by concept definitions. These measures are not absolute, yet they depend on the knowledge base they are applied to. Thus, the choice of the optimal feature sets may require a preliminary feature construction phase. To this purpose we have proposed randomized procedures based on *genetic programming* or *simulated annealing* [8, 9].

Regarding conceptual clustering, the expressiveness of the language adopted for describing objects and clusters is equally important. Former alternative methods devised for terminological representations, pursued logic-based approaches for specific languages [17, 10]. Besides of the language-dependency limitation, 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 should be more noise-tolerant and language-independent.

Thus 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. the novel measure. The notion of *means* characterizing partitional algorithms descending from (BISECTING) K-MEANS [15] originally developed for numeric or ordinal features, is replaced by the notion of *medoids* [16] as central individuals in a cluster. Hence we propose a BISECTING AROUND MEDOIDS algorithm, which exploits the aforementioned measures [8].

The clustering algorithm produces hierarchies of clusters. An evaluation of the method applied to real ontologies is presented based on internal validity indices such as the silhouette measure [16]. Then, we also suggest two possible ways for exploiting the outcomes of clustering: concept formation and detect concept drift or novelty detection. Namely, existing concept learning algorithms for DLs [14, 20] can be used to produce new concepts based on a group of examples (i.e. individuals in a cluster) and counterexamples (individuals in disjoint clusters, on the same hierarchy level). Besides, we provide also a method to detect interesting cases of concepts that are evolving or novel concepts which are emerging based on the elicited clusters.

The paper is organized as follows. Sect. 2 recalls the basics of the representation and the distance measure adopted. The clustering algorithm is presented and discussed in Sect. 3. After Sect. 4, concerning the related works, we present an experimental evaluation of the clustering procedure in Sect. 5. Conclusions are finally examined in Sect. 6.

2 Semantic Distance Measures

In the following, we assume that resources, concepts and their relationship may be defined in terms of a generic ontology representation 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.

2.1 A Semantic Semi-distance for Individuals

For our purposes, a function for measuring the similarity of individuals rather than concepts is needed. It can be observed that individuals do not have a syntactic structure that can be compared. This has led to lifting them to the concept description level before comparing them (recurring to the approximation of the *most specific concept* of an individual w.r.t. the ABox) [5].

We have developed a new measure whose definition totally depends on semantic aspects of the individuals in the knowledge base [8]. On a semantic level, similar individuals should behave similarly with respect to the same concepts. The computation of the similarity of individuals 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 [25], 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 individuals 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.

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

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 a set of concept descriptions $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, a family of functions*

$$d_p^{\mathsf{F}} : \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mapsto [0, 1]$$

is 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:*

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

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 $p > 0$, d_p is a semi-distance, i.e. given any three instances $a, b, c \in \mathsf{Ind}(\mathcal{A})$, it holds that:*

1. *$d_p(a, b) \geq 0$ and $d_p(a, b) = 0$ if $a = b$*
2. *$d_p(a, b) = d_p(b, a)$*
3. *$d_p(a, c) \leq d_p(a, b) + d_p(b, c)$*

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 [4], 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).

In the perspective of integrating the measure in ontology mining algorithms which massively use it, such as all instance-based methods, it should be noted that the π_i functions ($\forall i = 1, \ldots, m$) can be computed in advance for the training instances, thus determining a speed-up in the overall computation.

2.2 Feature Set Optimization

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 (see Sect. 5). However, the choice of the concepts to be included – *feature selection* – may be crucial, for a good committee may discern the individuals better and a possibly smaller committee yields more efficiency when computing the distance.

We have devised a specific optimization algorithm 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 [9]. Namely, since the function is very dependent on the concepts included in the committee of features F, two immediate heuristics can be derived: 1) control the number of concepts of the committee, including especially those that are endowed with a real discriminating power; 2) finding optimal sets of discriminating features, by allowing also their composition employing the specific constructors made available by the representation language of choice.

3 Hierarchical Clustering for Individuals in an Ontology

The conceptual clustering procedure that we propose can be ascribed to the category of the heuristic partitioning algorithms such as K-MEANS [15]. For the categorical nature of the assertions on individuals the notion of mean is replaced by the one of medoid, as in PAM (*Partition Around Medoids* [16]). Besides the procedure is crafted to work iteratively to produce a hierarchical clustering.

The algorithm implements a top-down bisecting method, starting with one universal cluster grouping all instances. Iteratively, it creates two new clusters by bisecting an existing one and this continues until the desired number of clusters is reached. This algorithm can be thought as levelwise producing a dendrogram: the number of levels coincides with the number of clusters.

Each cluster is represented by one of its individuals. As mentioned above, we consider the notion of medoid as representing a cluster center since our distance measure works on a categorical feature-space. The medoid of a group of individuals is the individual that has the lowest dissimilarity w.r.t. the others. Formally. given a cluster $C = \{a_1, a_2, \ldots, a_n\}$, the medoid is defined:

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

The proposed method can be considered as a hierarchical extension of PAM. A bipartition 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 final 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 [27]). This cluster is candidate to being splitted. The partition is constructed around two medoids initially chosen (SELECTMOSTDISSIMILAR() function) as the most dissimilar elements in the cluster and then iteratively adjusted in

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

begin
level $\leftarrow$ 0;
clusterVector[1] $\leftarrow$ allIndividuals;
repeat
        ++level;
        cluster2split $\leftarrow$ SELECTWORSTCLUSTER(clusterVector[level]);
        iterCount $\leftarrow$ 0;
        stableConfiguration $\leftarrow$ false;
        (newMedoid1,newMedoid2) $\leftarrow$ SELECTMOSTDISSIMILAR(cluster2split);
        repeat
            ++iterCount;
            (medoid1,medoid2) $\leftarrow$ (newMedoid1,newMedoid2);
            (cluster1,cluster2) $\leftarrow$ DISTRIBUTE(cluster2split,medoid1,medoid2);
            newMedoid1 $\leftarrow$ MEDOID(cluster1);
            newMedoid2 $\leftarrow$ MEDOID(cluster2);
            stableConfiguration $\leftarrow$ (medoid1 = newMedoid1) $\wedge$ (medoid2 = newMedoid2);
        until stableConfiguration $\vee$ (iterCount = maxIterations);
        clusterVector[level+1] $\leftarrow$ REPLACE(cluster2split,cluster1,cluster2,clusterVector[level]);
until (level $= k$);
end
```

Fig. 1. The HIERARCHICAL BISECTING AROUND MEDOIDS Algorithm

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 algorithm, where medoids are considered instead of means that can hardly be defined in symbolic computations. Then, the standard two steps are performed iteratively:

distribution given the current medoids, distribute the other individuals to either partition on the grounds of their distance w.r.t. the respective medoid;

medoid re-computation given the bipartition obtained by DISTRIBUTE(), compute the new medoids for either cluster.

The medoid tend to change at each iteration until eventually they converge to a stable couple (or when a maximum number of iterations have been performed).

An adaptation of 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.

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 less sensitive to the presence of outliers. In K-MEANS 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 more negatively affected by a single outlier.

3.1 Evolution: Automated Concept Drift and Novelty Detection

As mentioned in the introduction conceptual clustering enables a series of further activities related to dynamic settings: 1) concept drift [28]: i.e. the change of known concepts w.r.t. the evidence provided by new annotated individuals that may be made available over time; 2) novelty detection [26]: isolated clusters in the search space that require to be defined through new emerging concepts to be added to the knowledge base.

The algorithms presented above are suitable for an online unsupervised learning implementation. Indeed as soon as new annotated individuals are made available these may be assigned to the *closest* clusters (where closeness is measured as the distance to the cluster medoids or to the minimal distance to its instances). Then, new runs of the clustering algorithm may yield a modification of the original clustering both in the clusters composition and in their number.

Following [26], the clustering representing the starting concepts is built based on the clustering algorithm. For each cluster, the maximum distance between its instances and the medoid is computed. This establishes a decision boundary for each cluster. The union of the boundaries of all clusters represents a global decision boundary which defines the current model.

A new unseen example that falls inside this global boundary is consistent with the model and therefore considered *normal* and may be classified according to the current clustering; otherwise, a further analysis is needed. A single individual located externally w.r.t. the boundary should not be considered as novel *per se*, since it could somehow simply represent noise. Due to lack of evidence, these individuals are stored in a short-term memory, which is to be monitored for detecting the formation of new clusters that might indicate two conditions: novelty and concept drift. Namely, the clustering algorithm applied to individuals in the short-term memory generates candidate clusters. In order to validate a candidate cluster w.r.t. the current model (clustering), the algorithm in Fig. 2 can be applied.

The candidate cluster emerging from the short-memory candCluster is considered *abnormal*[2] (for the aims of drift or novelty detection) when the mean distance between

[2] This aims at choosing clusters whose density is not lower than that of the model.

```
(decision, newClustering) DRIFT_NOVELTY_DETECTION(currModel, candCluster)
input:  currModel: current clustering;
        candCluster: candidate cluster;
output: (decision, newClustering);

begin
$m_{\mathsf{CC}} := \mathrm{medoid}(\mathsf{candCluster})$;
for each $C_j \in \mathsf{currModel}$ do
        $m_j := \mathrm{medoid}(\mathsf{C_j})$;
overallAvgDistance $:= \frac{1}{|\mathsf{currModel}|} \sum_{C_j \in \mathsf{currModel}} \left[ \frac{1}{|C_j|} \sum_{a \in C_j} d(a, m_j) \right]$;
candClusterAvgDistance $:= \frac{1}{|\mathsf{candCluster}|} \sum_{a \in \mathsf{CCluster}} d(a, m_{\mathsf{CC}})$;
if overallAvgDistance $\geq$ candClusterAvgDistance then
        begin // abnormal candidate cluster detected
        $\overline{m} := \mathrm{medoid}(\{m_j \mid C_j \in \mathsf{currModel}\})$; // global medoid
        thrDistance $:= \max_{m_j \in \mathsf{currModel}} d(\overline{m}, m_j)$;
        if $d(\overline{m}, m_{\mathsf{CC}}) \leq$ thrDistance then
                return (drift, replace(currModel, candCluster))
        else
                return (novelty, currModel $\cup$ candCluster)
        end
else return (normal, integrate(currModel, candCluster))
end
```

Fig. 2. Concept drift and novelty detection algorithm

medoids and the respective instances, averaged over all clusters of the current model (overallAvgDistance) is greater than the average distance from the new individuals to the medoid of the candidate cluster (candClusterAvgDistance).

Then a threshold distance thrDistance for distinguishing between concept drift and novelty is computed as the maximum distance between the medoids of clusters in the original model m_j's and the global medoid[3] $\overline{m}$. When the distance between the overall medoid and the candidate cluster medoid m_{CC} does not exceed the threshold distance then a concept drift case is detected and the candidate cluster can replace an old cluster in the current clustering model. This may simply amount to reassigning the individuals in the drifted cluster to the new clusters or it may even involve a further run of the clustering algorithm to restructure the clustering model. Otherwise (novelty case) the clustering is simply extended with the addition of the new candidate cluster. Finally, when the candidate cluster is made up of normal instances these can be integrated by assigning them to the closest clusters.

The main differences from the original method [26], lie in the different representational setting (simple numeric tuples were considered) which allows for the use of off-the-shelf clustering methods such as k-MEANS [15] based on a notion of centroid

[3] Clusters which are closer to the boundaries of the model are more likely to appear due to a drift occurred in the normal concept. On the other hand, a candidate cluster appearing to be far from the normal concept may represent a novel concept.

which depend on the number of clusters required as a parameter. In our categorical setting, medoids substitute the role of means (or centroids) and, more importantly, our method is able to detect an optimal number of clusters autonomously, hence the influence of this parameter is reduced.

3.2 From Clusters to Concepts

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 solving 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) [1], such as $\mathcal{ALN}$, $\mathcal{ALE}$ or $\mathcal{ALC}$. This may involve the following steps:
given a cluster of individuals node_j

```
– for each individual $a_i \in \mathsf{node}_j$ do
        compute $M_i \leftarrow \mathsf{MSC}(a_i)$ w.r.t. $\mathcal{A}$;
– let $\mathsf{MSCs}_j \leftarrow \{M_i \mid a_i \in \mathsf{node}_j\}$;
– return $\mathsf{LCS}(\mathsf{MSCs}_j)$
```

However, the use of this generalizing operator may be criticized for the sensitiveness to the presence of outliers and for the excessive specificity of the resulting concepts which may result in poorly predictive descriptions w.r.t. future unseen individuals. yet this also depends on the degree of approximation of the MSC's.

As an alternative, algorithms for learning concept descriptions expressed in DLs may be employed [19, 14]. 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 these representations may be utilized for this new task.

4 Related Work

The unsupervised learning procedure presented in this paper is mainly based on two factors: the semantic dissimilarity measure and the clustering method. To the best of our knowledge in the literature there are very few examples of similar clustering algorithms working on complex representations that are suitable for knowledge bases of semantically annotated resources. Thus, in this section, we briefly discuss sources of inspiration for our procedure and some related approaches.

4.1 Relational Similarity Measures

As previously mentioned, 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 [4] or they are not completely semantic depending also on the structure of the descriptions [5]. Very few works deal with the comparison of individuals rather than concepts.

In the context of clausal logics, a metric was defined [23] for the Herbrand interpretations of logic clauses as induced from a distance defined on the space of ground atoms. This kind of measures may be employed to assess similarity in *deductive databases*. Although it represents a form of fully semantic measure, different assumptions are made with respect to those which are standard for knowledgeable bases in the SW perspective. Therefore the transposition to the context of interest is not straightforward.

Our measure is mainly based on Minkowski's measures [29] and on a method for distance induction developed by Sebag [25] in the context of *machine learning*, where *metric learning* is developing as an important subfield. In this work it is shown that the induced measure could be accurate when employed for classification tasks even though set of features to be used were not the optimal ones (or they were redundant). Indeed, differently from our unsupervised learning approach, the original method learns different versions of the same target concept, which are then employed in a voting procedure similar to the Nearest Neighbor approach for determining the classification of instances.

A source of inspiration was also *rough sets* theory [24] 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 Sect. 6) and suggest new ways to treat uncertainty.

4.2 Clustering Procedures

Our algorithm adapts to the specific representations devised for the SW context a combination of the distance-based approaches (see [15]). Specifically, in the methods derived from K-MEANS and K-MEDOIDS each cluster is represented by one of its points.

PAM, CLARA [16], and CLARANS [22] represent early systems adopting this approach. They implement iterative optimization methods that essentially cyclically relocate points between perspective clusters and recompute potential medoids. Ester et al. [7] extended CLARANS to deal with very large spatial databases.

Further comparable clustering methods are those based on an *indiscernibility relationship* [13]. 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 eventually induces clusters as equivalence classes.

Alternatively evolutionary clustering approaches may be considered [9] which are also capable to determine a good estimate of the number of clusters [11, 12]. The UNC algorithm is a more recent related approach which was also extended to the hierarchical clustering case H-UNC [21].

As mentioned in the introduction, the classic approaches to conceptual clustering [27] in complex (multi-relational) spaces are based on structure and logics. Kietz & Morik proposed a method for efficient construction of knowledge bases for the BACK representation language [17]. This method exploits the assertions concerning the roles available in the knowledge base, in order to assess, in the corresponding relationship,

Table 1. Ontologies employed in the experiments

ontology	DL	#concepts	#obj. prop.	#data prop.	#individuals
FSM	$\mathcal{SOF}(D)$	20	10	7	37
S.-W.-M.	$\mathcal{ALCOF}(D)$	19	9	1	115
Transportation	$\mathcal{ALC}$	44	7	0	250
Financial	$\mathcal{ALCIF}$	60	17	0	652
NTN	$\mathcal{SHIF}(D)$	47	27	8	676

those subgroups of the domain and ranges which may be inductively deemed as disjoint. In the successive phase, supervised learning methods are used on the discovered disjoint subgroups to construct new concepts that account for them. A similar approach is followed in [10], where the supervised phase is performed as an iterative refinement step, exploiting suitable refinement operators for a different DL, namely $\mathcal{ALC}$.

System OLINDDA [26] is one of the first methods exploiting clustering for detecting concept drift and novelty. Our method improves on it both in the representation of the instances and in being based on an original clustering method which is not parametrized on the number of clusters.

5 Experimental Evaluation of the Clustering Procedure

An experimental session was planned in order to prove the method feasible. It could not be a comparative experimentation since, to the best of our knowledge no other hierarchical clustering method has been proposed which is able to cope with DLs representations (on a semantic level) except [17, 10] which are language-dependent and produce non-hierarchical clusterings.

For the experiments, a number of different ontologies represented in OWL were selected, namely: FSM, Surface-Water-Model, Transportation and NewTestamentNames from the Protégé library[4], the Financial ontology[5] employed as a testbed for the Pellet reasoner. Table 1 summarizes important details concerning the ontologies employed in the experimentation.

A preliminary phase, may regard the selection of the features for the metric. Experimentally, we noted that the optimization affected the efficiency of the distance computation more than the metric sensitiveness. Thus we decided to employ the whole set of named concepts in the KB as features.

As pointed out in several surveys on clustering, it is better to use a different criterion for clustering (e.g. for choosing the candidate cluster to bisection) and for assessing the quality of a cluster. For the evaluation we employed standard validity measures for clustering: the mean square error (WSS, a measure of cohesion) and the *silhouette* measure [16]. Besides, we propose a the extension of Dunn's validity index for clusterings produced by the hierarchical algorithm [2]. Namely, we propose a modified version of

[4] `http://protege.stanford.edu/plugins/owl/owl-library`

[5] `http://www.cs.put.poznan.pl/alawrynowicz/financial.owl`

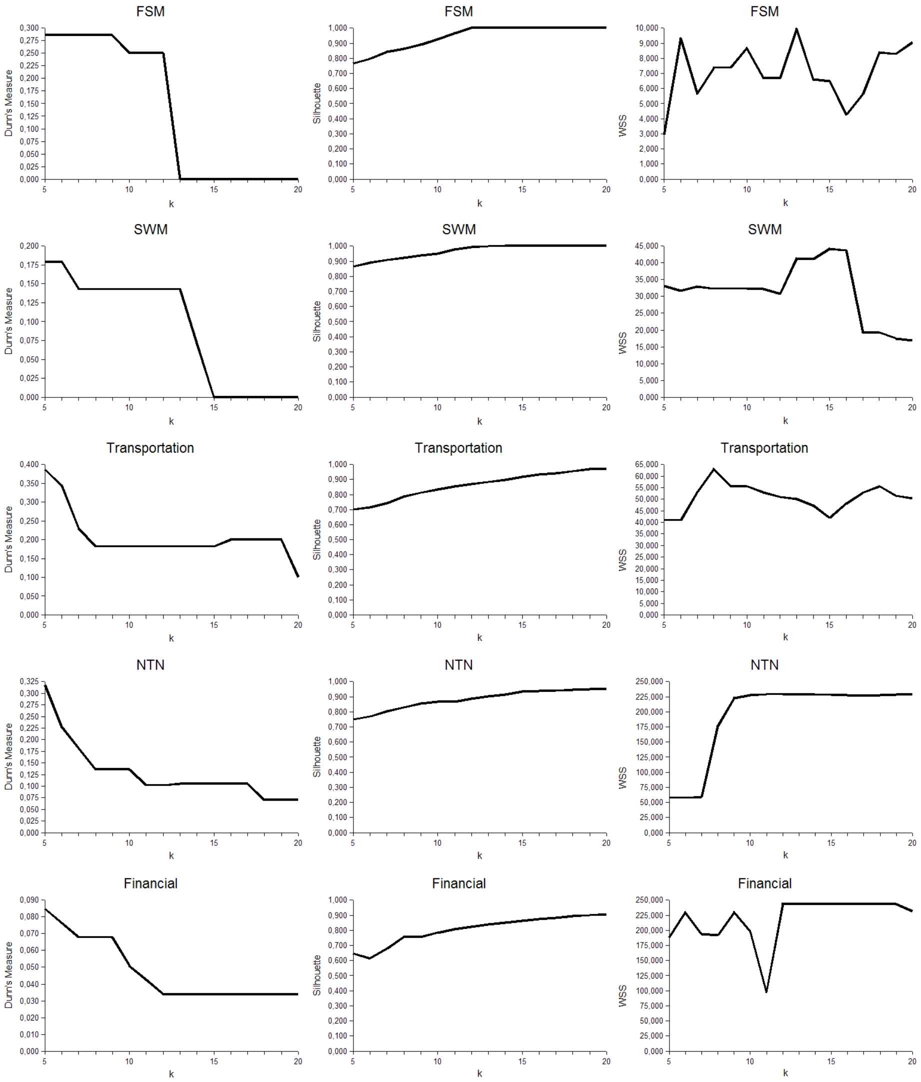

Fig. 3. Outcomes of the experiments: Dunn's, Silhouette, and WSS index graphs

Dunn's index to deal with medoids. Let $P = \{C_1, \ldots, C_k\}$ be a possible clustering of n individuals in k clusters. The index can be defined:

$$V_{GD}(P) = \min_{1 \leq i \leq k} \left\{ \min_{\substack{1 \leq j \leq k \\ i \neq j}} \left\{ \frac{\delta_p(C_i, C_j)}{\max_{1 \leq h \leq k} \{\Delta_p(C_h)\}} \right\} \right\}$$

where δ_p is the Hausdorff distance for clusters[6] derived from d_p and the cluster diameter measure Δ_p is defined:

$$\Delta_p(C_h) = \frac{2}{|C_h|}\left(\sum_{c \in C_h} d_p(c, m_h)\right)$$

which is more noise-tolerant w.r.t. other standard measures.

For each populated ontology, the experiments have been repeated for varying numbers k of clusters (5 through 20). In the computation of the distances between individuals (the most time-consuming operation) all concepts in the ontology have been used for the committee of features, thus guaranteeing meaningful measures with high redundancy. The PELLET reasoner[7] was employed to compute the projections. An overall experimentation of 16 repetitions on a dataset took from a few minutes to 1.5 hours on a 2.5GhZ (512Mb RAM) Linux Machine.

The outcomes of the experiments are reported in Fig. 3. For each ontology, we report the graph for Dunn's, Silhouette and WSS indexes, respectively, at increasing values of k (number of clusters searched, which determines the stopping condition).

Particularly, the decay of Dunn's index may be exploited as a hint on possible cut points (the *knees*) in the hierarchical clusterings (i.e. optimal values of k).

It is also possible to note that the silhouette values, as absolute clustering quality measures, are quite stably close to the top of the scale (1). This gives a way to assess the effectiveness of our algorithms w.r.t. others, although applied to different representations.

Conversely, the cohesion coefficient WSS may vary a lot, indicating that for some level the clustering found by the algorithm, which proceeds by bisection of the worst cluster in the previous level, is not the natural one, and thus is likely to be discarded.

6 Conclusions and Outlook

This work has presented a clustering method for the standard (multi-)relational representations adopted in the SW research. 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.

Currently we are working on extensions of the metric based on weighting the discernibility power of the features based on information theory (entropy). Besides, we are also investigating evolutionary clustering methods both for performing the optimization of the feature committee and for clustering individuals automatically discovering an optimal number of clusters [9].

[6] The metric δ_p is defined, given any couple of clusters (C_i, C_j), $\delta(C_i, C_j) = \max\{d_p(C_i, C_j), d_p(C_j, C_i)\}$, where $d_p(C_i, C_j) = \max_{a \in C_i}\{\min_{b \in C_j}\{d_p(a, b)\}\}$.

[7] http://pellet.owldl.com

Finally, we plan to perform further experiments to evaluate the quality of the clustering and of the induced (new) concepts, although it may be questionable to assess this objectively. The output of our method is thought to be validated by a domain expert. then, a knowledge engineer may foresee to adopt these methods to validate knowledge bases under construction while the experts are collaborating on the task.

References

[1] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook. Cambridge University Press, Cambridge (2003)

[2] Bezdek, J.C., Pal, N.R.: Some new indexes of cluster validity. IEEE Transactions on Systems, Man, and Cybernetics 28(3), 301–315 (1998)

[3] Borgida, A.: On the relative expressiveness of description logics and predicate logics. Artificial Intelligence 82(1-2)

[4] Borgida, A., Walsh, T.J., Hirsh, H.: Towards measuring similarity in description logics. In: Horrocks, I., Sattler, U., Wolter, F. (eds.) Working Notes of the International Description Logics Workshop, Edinburgh, UK. CEUR Workshop Proceedings, vol. 147 (2005)

[5] d'Amato, C., Fanizzi, N., Esposito, F.: Reasoning by analogy in description logics through instance-based learning. In: Tummarello, G., Bouquet, P., Signore, O. (eds.) Proceedings of Semantic Web Applications and Perspectives, 3rd Italian Semantic Web Workshop, SWAP 2006, Pisa, Italy. CEUR Workshop Proceedings, vol. 201 (2006)

[6] d'Amato, C., Staab, S., Fanizzi, N., Esposito, F.: Efficient discovery of services specified in description logics languages. In: Di Noia, T., et al. (eds.) Proceedings of Service Matchmaking and Resource Retrieval in the Semantic Web Workshop at ISWC 2007, vol. 243, CEUR (2007)

[7] Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases. In: Proceedings of the 2nd Conference of ACM SIGKDD, pp. 226–231 (1996)

[8] Fanizzi, N., d'Amato, C., Esposito, F.: Induction of optimal semi-distances for individuals based on feature sets. In: Working Notes of the International Description Logics Workshop, DL 2007, Bressanone, Italy. CEUR Workshop Proceedings, vol. 250 (2007)

[9] Fanizzi, N., d'Amato, C., Esposito, F.: Randomized metric induction and evolutionary conceptual clustering for semantic knowledge bases. In: Silva, M., Laender, A., Baeza-Yates, R., McGuinness, D., Olsen, O., Olstad, B. (eds.) Proceedings of the ACM International Conference on Knowledge Management, CIKM 2007, Lisbon, Portugal, ACM Press, New York (2007)

[10] Fanizzi, N., Iannone, L., Palmisano, I., Semeraro, G.: Concept formation in expressive description logics. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 99–113. Springer, Heidelberg (2004)

[11] Ghozeil, A., Fogel, D.B.: Discovering patterns in spatial data using evolutionary programming. In: Koza, J.R., Goldberg, D.E., Fogel, D.B., Riolo, R.L. (eds.) Genetic Programming 1996: Proceedings of the First Annual Conference, Stanford University, CA, USA, pp. 521–527. MIT Press, Cambridge (1996)

[12] Hall, L.O., Özyurt, I.B., Bezdek, J.C.: Clustering with a genetically optimized approach. IEEE Trans. Evolutionary Computation 3(2), 103–112 (1999)

[13] Hirano, S., Tsumoto, S.: An indiscernibility-based clustering method. In: Hu, X., Liu, Q., Skowron, A., Lin, T.Y., Yager, R., Zhang, B. (eds.) 2005 IEEE International Conference on Granular Computing, pp. 468–473. IEEE, Los Alamitos (2005)

[14] Iannone, L., Palmisano, I., Fanizzi, N.: An algorithm based on counterfactuals for concept learning in the semantic web. Applied Intelligence 26(2), 139–159 (2007)

[15] Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Computing Surveys 31(3), 264–323 (1999)

[16] Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, Chichester (1990)

[17] Kietz, J.-U., Morik, K.: A polynomial approach to the constructive induction of structural knowledge. Machine Learning 14(2), 193–218 (1994)

[18] Kirsten, M., Wrobel, S.: Relational distance-based clustering. In: Page, D.L. (ed.) ILP 1998. LNCS, vol. 1446, pp. 261–270. Springer, Heidelberg (1998)

[19] Lehmann, J.: Concept learning in description logics. Master's thesis, Dresden University of Technology (2006)

[20] Lehmann, J., Hitzler, P.: A refinement operator based learning algorithm for the alc description logic. In: The 17th International Conference on Inductive Logic Programming (ILP). LNCS, Springer, Heidelberg (2007)

[21] Nasraoui, O., Krishnapuram, R.: One step evolutionary mining of context sensitive associations and web navigation patterns. In: Proceedings of the SIAM conference on Data Mining, Arlington, VA, pp. 531–547 (2002)

[22] Ng, R., Han, J.: Efficient and effective clustering method for spatial data mining. In: Proceedings of the 20th Conference on Very Large Databases, VLDB 1994, pp. 144–155 (1994)

[23] Nienhuys-Cheng, S.-H.: Distances and limits on herbrand interpretations. In: Page, D.L. (ed.) ILP 1998. LNCS, vol. 1446, pp. 250–260. Springer, Heidelberg (1998)

[24] Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)

[25] Sebag, M.: Distance induction in first order logic. In: Džeroski, S., Lavrač, N. (eds.) ILP 1997. LNCS, vol. 1297, pp. 264–272. Springer, Heidelberg (1997)

[26] Spinosa, E.J., Ponce de Leon Ferreira de Carvalho, A., Gama, J.: OLINDDA: A cluster-based approach for detecting novelty and concept drift in data streams. In: Proceedings of the 22nd Annual ACM Symposium of Applied Computing, SAC 2007, Seoul, South Korea, vol. 1, pp. 448–452. ACM, New York (2007)

[27] Stepp, R.E., Michalski, R.S.: Conceptual clustering of structured objects: A goal-oriented approach. Artificial Intelligence 28(1), 43–69 (1986)

[28] Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23(1), 69–101 (1996)

[29] Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search – The Metric Space Approach. In: Advances in Database Systems, Springer, Heidelberg (2007)