Conceptual Clustering Applied to Ontologies

A Distance-based Evolutionary Approach

Floriana Esposito, Nicola Fanizzi, and Claudia d'Amato

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

Abstract. A clustering method is presented which can be applied to semantically annotated resources in the context of ontological knowledge bases. This method can be used to discover emerging groupings of resources expressed in the standard ontology languages. The method exploits a language-independent semi-distance measure over the space of resources, that is based on their semantics w.r.t. a number of dimensions corresponding to a committee of discriminating features represented by concept descriptions. A maximally discriminating group of features can be constructed through a feature construction method based on genetic programming. The evolutionary clustering algorithm proposed is based on the notion of medoids applied to relational representations. It is able to induce a set of clusters by means of a fitness function based on a discernibility criterion. An experimentation with some ontologies proves the feasibility of our method.

1 Introduction

In the perspective of the Semantic Web [2] knowledge bases will contain rich data and meta-data described with complex representations. This requires re-thinking the current data mining approaches to cope with the challenge of the new representation and semantics. In this work, unsupervised learning is tackled in the context of the standard concept languages used for representing ontologies which are based on *Description Logics* (henceforth DLs) [1]. In particular, we focus on the problem of *conceptual clustering* [25] for semantically annotated resources.

The benefits of clustering in the context of semantically annotated knowledge bases are manifold. Clustering enables the definition of new emerging categories (*concept formation*) on the grounds of the primitive concepts asserted in a knowledge base [9]; supervised methods can exploit these clusters to induce new concept definitions or to refining existing ones *ontology evolution*; intensionally defined groupings may speed-up the task of *discovery* and search in general.

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). Thus, clustering methods have aimed at defining groups of objects through conjunctive descriptions based on selected attributes [25].

Often these methods cannot into account any form of *prior knowledge* at a conceptual level encoding some semantic relationships. This hinders the interpretation of the outcomes of these methods which is crucial in the Semantic Web perspective in which the expressiveness of the language adopted for describing objects and clusters is extremely important. Specific logic-based approaches, intended for terminological representations [1], have have been proposed as language-dependent methods [16, 9]. These methods have been criticized for suffering from noise in the data. This motivates our investigation on similarity-based clustering approaches which can be more noise-tolerant, and as language-independent as possible. Specifically we propose a multi-relational extension of effective clustering techniques intended for grouping similar resources w.r.t. a semantic dissimilarity measure, which is tailored for the standard representations of Semantic Web context.

From a technical viewpoint, adapting existing algorithms to work on complex representations, requires semantic measures that are suitable for such concept languages. Recently, dissimilarity measures for specific DLs have been proposed [5]. Although they turned out to be quite effective for the inductive tasks, they were still partly based on structural criteria which makes them fail to fully capture the underlying semantics and hardly scale to any standard ontology language. As pointed out in a seminal paper on similarity measures for DLs [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 other tasks, a notion of similarity between *individuals* is required.

Therefore, we have devised a family of dissimilarity measures for semantically annotated resources, which can overcome the mentioned limitations [8]. Following the criterion of semantic discernibility of individuals, these measures are suitable for a wide range of concept languages since they are merely based on the discernibility of the input individuals with respect to a fixed committee of features represented by concept definitions. As such the new measures are not absolute, yet they depend on the knowledge base they are applied to. Thus, also the choice of the optimal feature sets deserves a preliminary feature construction phase, which may be performed by means of a randomized search procedure based on *genetic programming*, whose operators are borrowed from recent works on ontology evolution [13].

The clustering algorithm that we propose adopts an evolutionary learning approach for adapting classic distance-based clustering approaches, such as the K-MEANS [14]. In our setting, instead of the notion of *centroid* that characterizes algorithms originally developed for numeric or ordinal features, we recur to the notion of *medoids* [15] as central individuals in a cluster. The clustering problem is solved by considering populations made up of strings of medoids with different lengths. The medoids are computed according to the semantic measure induced with the methodology introduced above. On each generation, the strings in the current population are evolved by mutation and cross-over operators, which are also able to change the number of medoids. Thus, this algorithm is also able to autonomously suggest a promising number of clusters.

The paper is organized as follows. Sect. 2 presents the basics of the representation and the similarity measure adopted in the clustering algorithm. This algorithm is illustrated and discussed in Sect. 3. Related methods and distance measures are recalled in Sect. 4 then an experimental session applying the method on real ontologies is reported in Sect. 5. Conclusions and extensions are finally examined in Sect. 6.

2 Semantic Distance Measures

One of the advantages 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 open-world semantics (see the handbook [1] for a thorough reference).

In this context, a *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is made up of a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. $\mathcal{T}$ is a set of concept definitions. $\mathcal{A}$ contains assertions (ground facts) concerning individuals. The set of the individuals occurring in $\mathcal{A}$ will be denoted with $\mathsf{Ind}(\mathcal{A})$. The *unique names assumption* can be made for such individuals: each is assumed to be identified by its own URI.

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

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

We have developed new measures 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. 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 [24] 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 (dissimilarity 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*

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

defined as follows:
$\forall a, b \in \mathsf{Ind}(\mathcal{A})$

$$d_p^{\mathsf{F}}(a, b) := \frac{1}{m} \left(\sum_{i=1}^{m} \mid \pi_i(a) - \pi_i(b) \mid^p \right)^{1/p}$$

where $p > 0$ *and* $\forall i \in \{1, \ldots, m\}$ *the* projection function π_i *is defined by:*
$\forall c \in \mathsf{Ind}(\mathcal{A})$

$$\pi_i(c) = \begin{cases} 1 & \mathcal{K} \models F_i(c) \\ 0 & \mathcal{K} \models \neg F_i(c) \\ 1/2 & otherwise \end{cases} \tag{1}$$

The case of $\pi_i(c) = 1/2$ corresponds to the case when a reasoner cannot give the truth value for a certain membership query. This is due to the *Open World Assumption* (OWA) normally made in the descriptive semantics [1].

It can be proved that these functions have almost all standard properties of distances [8]:

Proposition 2.1 (semi-distance). *For a fixed feature set* F *and* $p > 0$ *the function* d_p^{F} *is a semi-distance.*

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).

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.

2.2 Committee Optimization

The choice of the concepts to be included in the committee – *feature selection* – may be crucial. Experimentally, it was observed that good results could be obtained by using the very set of both primitive and defined concepts found in the ontology. However, some ontologies define very large sets of concepts which make the task unfeasible. Thus, we have devised a specific optimization algorithms founded in *genetic programming* which are able to find optimal choices of discriminating concept committees.

Various optimizations of the measures can be foreseen as concerns its definition. Among the possible sets of features we will prefer those that are able to discriminate the individuals in the ABox.

Since the function is very dependent on the concepts included in the committee of features F, two immediate heuristics can be derived:

- limit the number of concepts of the committee, including especially those that are endowed with a real discriminating power;

- find sets of discriminating features, by allowing also their composition employing the specific constructors made available by the representation language of choice.

Both these objectives can be accomplished by means of randomized optimization techniques especially when knowledge bases with large sets of individuals are available. Namely, part of the entire data can be drawn in order to learn optimal F sets, in advance with respect to the successive usage for all other purposes.

Specifically, we experimented the usage of genetic programming for constructing optimal sets of features. Thus we devised the algorithm depicted in Fig. 1. Essentially the algorithm searches the space of all possible feature committees starting from an initial guess (determined by $\textsc{makeInitialFS}(\mathcal{K})$) based on the concepts (both primitive and defined) currently referenced in the knowledge base $\mathcal{K}$.

The outer loop gradually augments the cardinality of the candidate committees. It is repeated until the algorithm realizes that employing larger feature committees would not yield a better fitness value with respect to the best fitness recorded in the previous iteration (with fewer features).

The inner loop is repeated for a number of generations until a stop criterion is met, based on the maximal value of generations $\mathsf{maxGenerations}$ or, alternatively, when an minimal threshold for the fitness value $\mathsf{minFitness}$ is reached by some feature set in the population, which can be returned.

As regards the $\textsc{bestFitness}()$ routine, it computes the best feature committee in a vector in terms of their *discernibility* [22, 12]. For instance, given the whole set of individuals $IS = \mathsf{Ind}(\mathcal{A})$ (or just a sample to be used to induce an optimal measure) the fitness function may be:

$$\textsc{discernibility}(\mathsf{F}) := \frac{1}{|IS|^2} \sum_{(a,b)\in IS^2} \sum_{i=1}^{|\mathsf{F}|} \frac{|\ \pi_i(a) - \pi_i(b)\ |}{2 \cdot |\mathsf{F}|}$$

As concerns finding candidate sets of concepts to replace the current committee ($\textsc{generateOffsprings}()$ routine), the function was implemented by recurring to simple transformations of a feature set:

- choose $\mathsf{F} \in \mathsf{currentFSs}$;
- randomly select $F_i \in \mathsf{F}$;
  - replace F_i with $F_i' \in \textsc{randomMutation}(F_i)$ randomly constructed, or
  - replace F_i with one of its refinements $F_i' \in \textsc{ref}(F_i)$

Refinement of concept description may be language specific. E.g. for the case of $\mathcal{ALC}$ logic, refinement operators have been proposed in [13].

This is iterated till a suitable number of offsprings is generated. Then these offspring feature sets are evaluated and the best ones are included in the new version of the $\mathsf{currentFSs}$ array; the minimal fitness value for these feature sets is also computed. As mentioned, when the while-loop is over the current best fitness is compared with the best one computed for the former feature set length; if an improvement is detected then the outer repeat-loop is continued, otherwise (one of) the former best feature set(s) is selected for being returned as the result of the algorithm.

```
FeatureSet OPTIMIZEFS($\mathcal{K}$, maxGenerations, minFitness)
input:
    $\mathcal{K}$: current knowledge base
    maxGenerations: maximal number of generations
    minFitness: minimal fitness value
output:
    FeatureSet: FeatureSet
begin
currentBestFitness := 0; formerBestFitness := 0;
currentFSs := MAKEINITIALFS($\mathcal{K}$); formerFSs := currentFSs;
repeat
    fitnessImproved := false;
    generationNumber := 0;
    currentBestFitness := BESTFITNESS(currentFSs);
    while (currentBestFitness $<$ minFitness) or (generationNumber $<$ maxGenerations)
        begin
        offsprings := GENERATEOFFSPRINGS(currentFSs);
        currentFSs := SELECTFROMPOPULATION(offsprings);
        currentBestFitness := BESTFITNESS(currentFSs);
        ++generationNumber;
        end
    if  (currentBestFitness $>$ formerBestFitness) and (currentBestFitness $<$ minFitness) then
        begin
        formerFSs := currentFSs;
        formerBestFitness := currentBestFitness;
        currentFSs := ENLARGEFS(currentFSs);
        end
    else fitnessImproved := true;
    end
until not fitnessImproved;
return BEST(formerFSs);
end
```

Fig. 1. Feature set optimization algorithm based on Genetic Programming.

Further methods for performing feature construction by means of randomized approaches are discussed in [8], where we propose a different approach based on *simulated annealing* in a DL framework, employing similar refinement operators.

3 Evolutionary Clustering Around Medoids

The conceptual clustering procedure consists of two phases: one that detects the clusters in the data and the other that finds an intensional definition for the groups of individuals detected in the former phase.

The first clustering phase implements a genetic programming learning scheme, where the designed representation for the competing genes is made up of strings (lists) of individuals of different lengths, where each individual stands as prototypical for one cluster. Thus, each cluster will be represented by its prototype recurring to the notion of *medoid* [15, 14] on a categorical feature-space w.r.t. the distance measure previously defined. Namely, 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:

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

The algorithm performs a search in the space of possible clusterings of the individuals optimizing a fitness measure maximizing discernibility of the individuals of the different clusters (inter-cluster separation) and the intra-cluster similarity measured in terms of our metric.

The second phase is more language dependent. The various cluster can be considered as training examples for a supervised algorithm aimed at finding an intensional DL definition for one cluster against the counterexamples, represented by individuals in different clusters [16, 9].

3.1 The Clustering Algorithm

The proposed clustering algorithm can be considered as an extension of methods based on genetic programming, where the notion of cluster prototypical instance of centroid, typical of the numeric feature-vector data representations, is replaced by that of medoid [15] as in (*Partition Around Medoids* or *PAM*): 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. In the algorithm, a genome will be represented by a list of medoids $G = \{m_1, \ldots, m_k\}$. Per each generation those that are considered as best w.r.t. a fitness function are selected for passing to the next generation. Note that the algorithm does not prescribe a fixed length of these lists (as, for instance in K-MEANS and its extensions [14]), hence it should be able to detect an optimal number of clusters for the data at hand.

Fig. 2 reports a sketch of the clustering algorithm. After the call to the initialization procedure INITIALIZE() returning the randomly generated initial population of medoid strings (currentPopulation) in a number of popLength, it essentially consists of the typical generation loop of genetic programming.

At each iteration this computes the new offsprings of current best clusterings represented by currentPopulation. This is performed by suitable genetic operators explained in the following. The fitnessVector recording the quality of the various offsprings (i.e. clusterings) is then updated, which is used to select the best offsprings that survive, passing to the next generation.

The quality of a genome $G = \{m_1, \ldots, m_k\}$ is evaluated by distributing all individuals among the clusters ideally formed around the medoids listed in it. Let C_i be the cluster around medoid m_i, $i = 1, \ldots, k$. Then, the measure is computed as follows:

$$\text{UNFITNESS}(G) := \sqrt{k+1} \sum_{i=1}^{k} \sum_{x \in C_i} d_p(x, m_i)$$

```
medoidVector ECM(maxGenerations, minGap)
input:
    maxGenerations: max number of iterations;
    minGap: minimal gap for stopping the evolution;
output:
    medoidVector: list of medoids
begin
INITIALIZE(currentPopulation,popLength);
while (generation $\leq$ maxGenerations) and (gap $>$ minGap)
    begin
    offsprings := GENERATEOFFSPRINGS(currentPopulation);
    fitnessVector := COMPUTEFITNESS(offsprings);
    currentPopulation := SELECT(offsprings,fitnessVector);
    gap := (UNFITNESS[popLength]$-$UNFITNESS[1]);
    generation++;
    end
return currentPopulation[0]; // best genome
end
```

Fig. 2. ECM: the EVOLUTIONARY CLUSTERING AROUND MEDOIDS algorithm.

This measure is to be minimized. The factor $\sqrt{k+1}$ is introduced in order to penalize those clusterings made up of too many clusters that could enforce the minimization in this way (e.g. by proliferating singletons). This can be considered a measure of incoherence *within* the various clusters, while the fitness function used in the metric optimization procedure measures discernibility as the spread of the various individuals in the derived space independently of their classification.

The loop condition is controlled by two factors the maximal number of generation (the maxGenerations parameter) and the difference (gap) between the fitness of best and of the worst selected genomes in currentPopulation (which is supposed to be sorted in ascending order, 1 through popLength). Thus another stopping criterion is met when this gap becomes less than the minimal gap minGap passed as a parameter to the algorithm, meaning that the algorithm has reached a (local) minimum.

It remains to specify the nature of the GENERATEOFFSPRINGS procedure function and the number of such offsprings, which may as well be another parameter of the ECM algorithm. Three mutation and one crossover operators are implemented:

DELETION(G) drop a randomly selected medoid:
$G := G \setminus \{m\}, m \in G$

INSERTION(G) select $m \in \mathsf{Ind}(\mathcal{A}) \setminus G$ that is added to G:
$G := G \cup \{m\}$

REPLACEMENTWITHNEIGHBOR(G) randomly select $m \in G$ and replace it with $m' \in \mathsf{Ind}(\mathcal{A}) \setminus G$ such that $\forall m'' \in \mathsf{Ind}(\mathcal{A}) \setminus G \; d(m, m') \leq d(m, m'')$:
$G' := (G \setminus \{m\}) \cup \{m'\}$

CROSSOVER(G_A,G_B) select subsets $S_A \subset G_A$ and $S_B \subset G_B$ and exchange them between the genomes:
$G_A := (G_A \setminus S_A) \cup S_B$ and $G_B := (G_B \setminus S_B) \cup S_A$

```
input   Clustering $= \{C_j \mid j = 1, \ldots, k\}$: set of clusters
        $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$: knowledge base;
output  Descriptions: set of DL concept descriptions
Descriptions := $\emptyset$;
for each $C_j \in$ Clustering:
    for each individual $a_i \in C_j$:
        do compute $M_i$ := msc$(a_i)$ w.r.t. $\mathcal{A}$;
    let MSCs$_j$ := $\{M_i \mid a_i \in C_j\}$;
    Descriptions := Descriptions $\cup\{$lcs(MSCs$_j)\}$;
return Descriptions;
```

Fig. 3. A basic concept induction algorithm from clusterings.

A (10+60) selection strategy has been implemented, indicating, resp., the number of parents selected for survival and the number of their offsprings.

3.2 The Supervised Learning Phase

Each cluster may be labeled with a new DL concept definition which characterizes the individuals in the given cluster while discriminating those in other clusters [9]. The process of labeling clusters 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, for DLs that allow for the computation of (an approximation of) the *most specific concept* (msc) and *least common subsumer* (lcs) [1] (such as $\mathcal{ALC}$) is depicted in Fig. 3.

However, such a solution is likely to produce overly specific definitions which may lack of predictiveness w.r.t. future individuals. Hence, better generalizing operators would be needed. Alternatively, algorithms for learning concept descriptions expressed in DLs may be employed [13]. Further refinement operators for the $\mathcal{ALC}$ DL have been proposed [18] to be employed in an algorithm performing a heuristic search in the refinement tree guided by a fitness function.

3.3 Discussion

For an analysis of the algorithm, the parameters of the methods based on genetic programming have to be considered, namely maximum number of iterations, number of offsprings, number of genomes that are selected for the next generation. However, it should be also pointed out that computing the fitness function requires some inference service (instance-checking) from a reasoner whose complexity may dominate the overall complexity of the process. This depends on the DL language of choice and also on the structure of the concepts descriptions handled, as investigated in the specific area (see [1], Ch. 3).

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. Density based methods could be also investigated, yet this may be difficult when handling complex data. 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.

Together with the density based clustering methods, also the algorithms based on medoids have several favorable properties w.r.t. other methods based on (dis)similarity. Since it performs clustering with respect to any specified metric, it allows for a flexible definition of the similarity function. 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: mostly they are rather based on a distances in Euclidean spaces. In addition, the algorithm has the advantage of identifying clusters by the medoids which represent more robust representations of the cluster centers that are less sensitive to outliers than other cluster profiles, such as the cluster centers 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 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.

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]. OSS is another recent proposal for an asymmetric similarity function for concepts within an ontology [23] based on its structure. Very few works deal with the comparison of individuals rather than concepts.

In the context of clausal logics, a metric was defined [21] 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 [26] and on a method for distance induction developed by Sebag [24] 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 [22] 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.

Our algorithm adapts to the specific representations devised for the SW context a combination of evolutionary clustering and the distance-based approaches (see [14]). Specifically, in the methods derived from K-MEANS and K-MEDOIDS each cluster is represented by one of its points.

Early versions of this approach are represented by further algorithms based on PAM such as CLARA [15], and CLARANS [20]. They implement iterative optimization methods that essentially cyclically relocate points between perspective clusters and recompute potential medoids. The leading principle for the process is the effect on an objective function. The whole dataset is assigned to resulting medoids, the objective function is computed, and the best system of medoids is retained. In CLARANS a graph is considered whose nodes are sets of k medoids and an edge connects two nodes if they differ by one medoid. While CLARA compares very few neighbors (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. Ester et al. [6] extended CLARANS to deal with very large spatial databases.

Our algorithm may be considered an extension of evolutionary clustering methods [11] which are also capable to determine a good estimate of the number of clusters [10]. Besides, we adopted the idea of representing clusterings (genomes) as strings of cluster centers [17] transposed to the case of medoids for the categorical search spaces of interest.

Other related recent approaches are represented by the UNC algorithm and its extension to the hierarchical clustering case H-UNC [19]. Essentially, UNC solves a multimodal function optimization problem seeking dense areas in the feature space. It is also able to determine their number. The algorithm is also demonstrated to be noise-tolerant and robust w.r.t. the presence of outliers. However, the applicability is limited to simpler representations w.r.t. those considered in this paper.

Further comparable clustering methods are those based on an *indiscernibility relationship* [12]. 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.

As mentioned in the introduction, the classic approaches to conceptual clustering [25] 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 [16]. This method exploits the assertions concerning the roles available in the knowledge base, in order to assess, in the corresponding relationship, 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 [9], where the supervised phase is performed as an iterative refinement step, exploiting suitable refinement operators for a different DL, namely $\mathcal{ALC}$.

5 Experimental Evaluation

A comparative evaluation of the method is not possible yet, since to the best of our knowledge, there is no similar algorithm which can cope with complex DL languages such as those indicated in the following Tab. 1. The only comparable (logical) approaches to clustering DL KBs are suitable for limited languages only (e.g. see [16, 9]).

The clustering procedure was validated through some standard internal indices [14, 3]. As pointed out in several surveys on clustering, it is better to use a different criterion for the clustering algorithm (e.g. for choosing the candidate cluster to bisection) and for assessing the quality of its resulting clusters.

To this purpose, we propose a generalization of Dunn's index [3] 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 derived from d_p (defined: $\delta_p(C_i, C_j) = \max\{d_p(C_i, C_j), d_p(C_j, C_i)\}$, where $d_p(C_i, C_j) = \max_{a \in C_i}\{\min_{b \in C_j}\{d_p(a, b)\}\}$) while the cluster diameter measure Δ_p is defined:

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

The other indices employed are more standard: the mean within-cluster square sum error (WSS), a measure of cohesion, and the silhouette measure [15].

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[1], the FINANCIAL ontology[2] employed as a testbed for the PELLET reasoner. Tab. 1 summarizes important details concerning the

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

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

Table 1. Ontologies employed in the experiments.

ONTOLOGY	DL	*#concepts*	*#object 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
NTN	$\mathcal{SHIF}(D)$	47	27	8	676
FINANCIAL	$\mathcal{ALCIF}$	60	16	0	1000

Table 2. Results of the experiments: average value (±std. deviation) and min−max value ranges.

ONTOLOGY	SILHOUETTE index	DUNN'S index	WSS index
FSM	.998 (±.005) .985−1.000	.221 (±.003) .212−.222	30.254 (±11.394) 14.344−41.724
S.-W.-M.	1.000 (±.000) 1.000−1.000	.333 (±.000) .333−.333	11.971 (±11.394) 7.335−13.554
TRANSPORTATION	.976 (±.000) .976−.976	.079 (±.000) .079−.079	46.812 (±5.944) 39.584−57.225
NTN	.986 (±.007) .974−.996	.058 (±.003) .056−.063	96.155 (±24.992) 64.756−143.895
FINANCIAL	.927 (±.034) .861−.951	.237 (±.000) .237−.237	130.863 (±24.117) 99.305−163.259

ontologies employed in the experimentation. A variable number of assertions per single individual was available in the ontology. For each ontology, the experiments have been repeated for 10 times. The PELLET 1.4 reasoner was employed to compute the projections required for determining the distance between individuals. An overall experimentation (10 repetitions) on a single ontology took from a few minutes up to less than one hour on a 2.5GhZ (512Mb RAM) Linux Machine.

The outcomes of the experiments are reported in Tab. 2. It is possible to note that the the Silhouette measure is quite close its optimal value (1), thus providing an absolute indication for the quality of the obtained clusterings. The variability is limited thus the performance appears to be quite stable.

Dunn's and WSS indices may be employed as a suggestion on whether to accept or not the (number of) clusters computed by the algorithm. Namely, among the various repetitions, those final clusterings whose values maximize these indices would have to be preferred. The high variance observed for the WSS index (that it is not limited within a range) has to be considered in proportion with its mean values. Besides, this measure is very sensitive to the number of clusters produced by the method. Although the algorithm converges to a stable number of clusters a difference of 1 may yield a sensible variation of the WSS, also because medoids are considered as centers rather than centroids.

6 Conclusions and Future Work

This work has presented a clustering for (multi-)relational representations which are standard in the Semantic Web 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 clustering procedures employing medoids since complex representations typical of the ontology in the Semantic Web are to be dealt with.

Better fitness functions may be investigated for both the evolutionary distance optimization procedure and the clustering one. In particular, feature selection for inducing a good distance measure deserves an independent investigation in order to make the choice efficient despite the large extent of the search space. As mentioned, we are investigating other stochastic procedures based on local search [8] and also extensions which can treat less uniformly the cases of uncertainty, e.g. evidence combination methods related to rough sets theory.

We are also devising extensions that are able to produce hierarchical clusterings [7] which would suggest new (non necessarily disjoint) concepts. Instead of repeatedly bisecting the target cluster (as in BISECTING K-MEANS [14]) the algorithm would autonomously find an optimal number for the split at each level.

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May 2001.

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

[4] A. Borgida, T.J. 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.

[5] 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.

[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, C. d'Amato, and F. Esposito. A hierarchical clustering procedure for semantically annotated resources. In R. Basili and M.T. Pazienza, editors, *Proceedings of the 10th Congress of the Italian Association for Artificial Intelligence, AI*IA2007*, volume 4733 of *LNAI*, pages 266–277. Springer, 2007.

[8] N. Fanizzi, C. d'Amato, and F. Esposito. Induction of optimal semi-distances for individuals based on feature sets. In *Working Notes of the International Description Logics Workshop, DL2007*, volume 250 of *CEUR Workshop Proceedings*, Bressanone, Italy, 2007.

[9] 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.

[10] A. Ghozeil and D.B. Fogel. Discovering patterns in spatial data using evolutionary programming. In John R. Koza, David E. Goldberg, David B. Fogel, and Rick L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 521–527, Stanford University, CA, USA, 1996. MIT Press.

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

[12] 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.

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

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

[15] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

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

[17] C.-Y. Lee and E. K. Antonsson. Variable length genomes for evolutionary algorithms. In L. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and H.-G. Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO00*, page 806. Morgan Kaufmann, 2000.

[18] J. Lehmann and P. Hitzler. A refinement operator based learning algorithm for the $\mathcal{ALC}$ description logic. In *Proceedings of the 17th International Conference on Inductive Logic Programming, ILP2007*, 2007.

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

[20] 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.

[21] 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.

[22] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, 1991.

[23] V. Schickel-Zuber and B. Faltings. OSS: A semantic similarity function based on hierarchical ontologies. In Manuela M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI2007*, pages 551–556, Hyderabad, India, 2007.

[24] 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.

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

[26] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search: The Metric Space Approach*. Springer, 2007.