Conceptual Clustering Applied to Ontologies by means of Semantic Discernability

Floriana Esposito, Nicola Fanizzi, and Claudia d'Amato

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 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 Introduction

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 automation 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* [12] in the context of semantically annotated knowledge bases are manifold. Clustering annotated resources enables the definition of new emerging concepts (*concept formation*) on the grounds of the primitive concepts asserted in a knowledge base; 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 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, conceptual clustering methods have aimed at defining groups of objects through conjunctive descriptions based on selected attributes [12]. In the perspective, the expressiveness of the language adopted for describing objects and clusters (concepts) is equally important. Alternative approaches, for terminological representations, pursued a different way for attacking the problem, devising logic-based methods for specific languages [9, 5]. 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, and as language-independent as possible. 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.

In this setting, instead of the notion of *means* that characterizes the algorithms descending from (BISECTING) K-MEANS [7] originally developed for numeric or ordinal features, we recur to the notion of *medoids* (like in PAM [8]) as central individuals in a cluster. Another theoretical problem is posed by the *Open World Assumption* (OWA) that is generally made on the language semantics, differently from the *Closed World Assumption* (CWA) which is standard in machine learning or query-answering. As pointed out in a seminal paper on similarity measures for DLs [3], 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.

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 makes them fail to fully grasp the underlying semantics and hardly scale to any standard ontology language. Therefore, we have devised a family of dissimilarity measures for semantically annotated resources, which can overcome the aforementioned limitations. Following the criterion of semantic discernibility of individuals, we present a new family of measures that is suitable a wide range of languages since it is 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 *simulated annealing*.

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. Conclusions are finally examined in Sect. 5.

2 Semantic Distance Measures

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.

2.1 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).

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 [11] 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*

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

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

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:*

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

2.2 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. and 2. are trivial. As for 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.

2.3 Feature Set Optimization

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.

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:

Definition 2.2 (good feature set). *Let* $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$ *be a set of concept descriptions. We call* F *a* good feature set *for the knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ *iff* $\forall a, b \in \mathsf{Ind}(\mathcal{A})\ \exists i \in \{1, \ldots, m\} :\ \pi_i(a) \neq \pi_i(b)$.

Then, when the function defined above is parameterized on a good feature set, it has the properties of a distance function.

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.

Both these objectives can be accomplished by means of machine learning 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.

We have been experimenting the usage of genetic programming for constructing an optimal set of features. Yet these are known to suffer from being possibly caught in local minima. An alternative is employing a different probabilistic search procedure which aims at a global optimization. Thus we devised a simulated annealing search, whose algorithm is depicted in Fig. 1.

Essentially the algorithm searches the space of all possible feature committees starting from an initial guess (determined by MAKEINITIALFS($\mathcal{K}$)) based on the concepts (both primitive and defined) currently referenced in the knowledge base. The loop controlling the search is repeated for a number of times that depends on the temperature which gradually decays to 0, when the current committee can be returned. The current feature set is iteratively refined calling a suitable procedure RANDOMSUCCESSOR(). Then the fitness of the new feature set is compared to that of the previous one determining the increment of energy ΔE. If this is non-null then the computed committee replaces the old one. Otherwise it will be replaced with a probability that depends on ΔE.

As regards the FITNESSVALUE(F), it can be computed as the *discernibility factor* of the feature set. For example given the whole set of individuals $IS = \mathsf{Ind}(\mathcal{A})$ (or just

FeatureSet OPTIMIZEFEATURESET($\mathcal{K}$, ΔT)
input $\mathcal{K}$: Knowledge base
ΔT: function controlling the decrease of temperature
output FeatureSet
static currentFS: current Feature Set
nextFS: next Feature Set
Temperature: controlling the probability of downward steps
begin
currentFS $\leftarrow$ MAKEINITIALFS($\mathcal{K}$)
for $t \leftarrow 1$ **to** ∞ **do**
Temperature $\leftarrow$ Temperature $- \Delta T(t)$
if (Temperature = 0)
return currentFS
nextFS $\leftarrow$ RANDOMSUCCESSOR(currentFS,$\mathcal{K}$)
$\Delta E \leftarrow$ FITNESSVALUE(nextFS) $-$ FITNESSVALUE(currentFS)
if ($\Delta E > 0$)
currentFS $\leftarrow$ nextFS
else *// replace FS with given probability*
REPLACE(currentFS, nextFS, $e^{\Delta E}$)
end

Fig. 1. Feature Set optimization based on a Simulated Annealing procedure.

a sample to be used to induce an optimal measure) the fitness function may be:

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

As concerns finding candidates to replace the current committee (RANDOMSUCCESSOR()), the function was implemented by recurring to simple transformations of a feature set:

- adding (resp. removing) a concept C: nextFS $\leftarrow$ currentFS $\cup \{C\}$
  (resp. nextFS $\leftarrow$ currentFS $\setminus \{C\}$)
- randomly choosing one of the current concepts from currentFS, say C;
  replacing it with one of its refinements $C' \in \textsc{ref}(C)$

Refinement of concept description is language specific. E.g. for the case of $\mathcal{ALC}$ logic, refinement operators have been proposed in [10, 6].

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 [7]. Each cluster is represented by its center. In our setting we will 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 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 proposed algorithm can be considered as a hierarchical extension of the PAM algorithm (*Partition Around Medoids* [8]): 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. 2 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 [12]). 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 algorithm, where medoids are considered instead of means that can hardly be defined in symbolic computations. Then, the classical two steps are performed in an iteration:

distribution 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;

center re-computation 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).

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.

```
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 := 0;
clusterVector[1] := allIndividuals;
repeat
        ++level;
        cluster2split := SELECTWORSTCLUSTER(clusterVector[level]);
        iterCount := 0;
        stableConfiguration := false;
        (newMedoid1,newMedoid2) := SELECTMOSTDISSIMILAR(cluster2split);
        repeat
            ++iterCount;
            (medoid1,medoid2) := (newMedoid1,newMedoid2);
            (cluster1,cluster2) := DISTRIBUTE(cluster2split,medoid1,medoid2);
            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);
end
```

Fig. 2. The HIERARCHICAL BISECTING AROUND MEDOIDS Algorithm.

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 $\mathcal{ALC}$). 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 [10, 6]. 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 Experimental Validation

We propose also a validation index that can be employed to assess the best clustering level in the dendrogram produced by the hierarchical algorithm. 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.

To this purpose, we modify a generalization of Dunn's index [2] 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[2] 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.

[2] 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 $d_p(C_i, C_j) = \max_{a \in C_i}\{\min_{b \in C_j}\{d_p(a, b)\}\}$.

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

The other measures employed are more standard: the mean square error (WSS, a measure of cohesion) and the silhouette measure [8].

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[3], the FINANCIAL ontology[4] employed as a testbed for the PELLET reasoner. Table 1 summarizes important details concerning the ontologies employed in the experimentation.

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 redundance. The PELLET reasoner[5] 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. It is possible to note that the two principal measures (Dunn's and Silhouette) are quite close to their optimal values (0 and 1, resp.), while the cohesion coefficient WSS may vary a lot, which gives an hint on possible cut points in the hierarchical clusterings (i.e. optimal values of k).

5 Conclusions

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.

Currently we are investigating evolutionary clustering methods both for performing the optimization of the feature committee and for clustering individuals automatically discovering an optimal number of clusters.

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

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

[5] `http://pellet.owldl.com`

References

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

[2] 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.

[3] 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.

[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] 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.

[6] 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.

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

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

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

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

[11] 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.

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

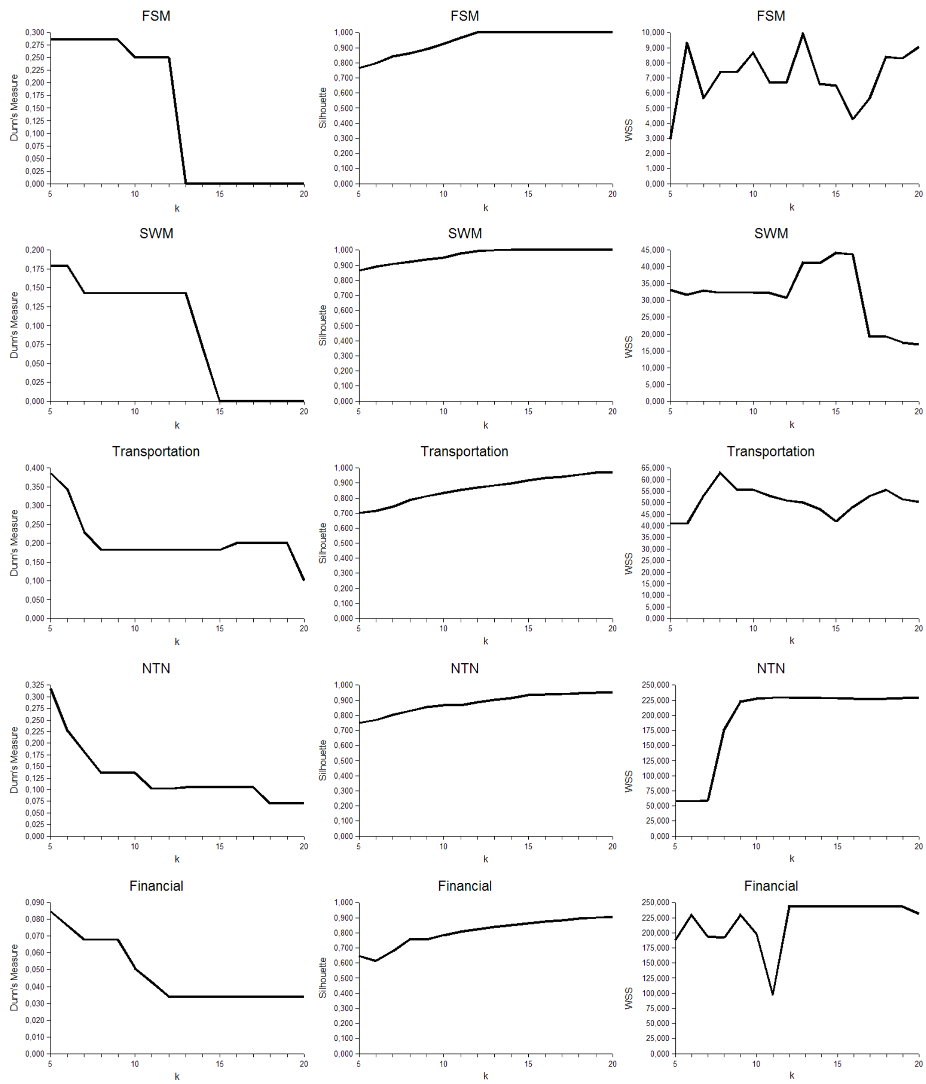

Fig. 3. Outcomes of the experiments: Dunn's, Silhouette, and WSS index graphs.