Learning with Kernels in Description Logics

Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito

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

Abstract. We tackle the problem of statistical learning in the standard knowledge base representations for the Semantic Web which are ultimately expressed in description Logics. Specifically, in our method a kernel functions for the $\mathcal{ALCN}$ logic integrates with a support vector machine which enables the usage of statistical learning with reference representations. Experiments where performed in which kernel classification is applied to the tasks of resource retrieval and query answering on OWL ontologies.

1 Learning from Ontologies

The Semantic Web (SW) represents an emerging applicative domain where knowledge intensive automated manipulations on complex relational descriptions are foreseen. Although machine learning techniques may have a great potential in this field, so far research has focused mainly on methods for knowledge acquisition from text (*ontology learning*) [4]. Yet machine learning methods can be transposed from ILP to be applied to ontologies described with formal concept representations employed to model knowledge bases in the SW.

Description Logics (DLs) [1] is a family of languages that has been adopted as the core technology for representing ontologies. Such languages are endowed with an open-world semantics which makes them particularly fit for the SW applications where, differently from classical DB contexts, knowledge bases are considered inherently incomplete, since new resources may continuously made available across the Web. Thus, few methods have been proposed for learning these representations (e.g. see [5, 13, 8, 14]).

While classic ILP techniques have been adapted to work with DLs representations, purely logic approaches often fall short in terms of efficiency and noise-tolerance. Learning with logic-based methods is inherently intractable in multi-relational settings. Moreover, for the sake of tractability, only very simple DL languages have been considered so far [6]. Recently, it has been shown that kernel methods may be effectively applied to structured representations [10] and also to ontology languages [9, 2].

In this work, a family of kernel functions is defined for DLs representations. Specifically, the $\mathcal{ALCN}$ logic [1] is adopted as a tradeoff between efficiency and expressiveness. The kernel functions are defined encoding a notion of similarity between objects expressed in this representation, which is based on structural and semantic aspects of the reference language, namely a normal form for the concept descriptions and the extension of concepts approximated through the objects that are (directly or provably) known to belong to them.

By coupling the kernel functions with support vector machines (SVMs) many tasks can be tackled. Particularly, we demonstrate how to perform important inference services based on inductive classification, namely concept retrieval and approximate query answering [1], that may turn out to be hard for logic methods, especially with knowledge bases built from heterogeneous sources. These tasks are generally grounded on merely deductive procedures which easily fail in case of (partially) inconsistent or incomplete knowledge. We show how inductive methods perform comparably well w.r.t. a standard deductive reasoner, allowing the suggestion of new knowledge that is not logically derivable similarly to *abductive* conclusions.

An experimentation on both artificial and real ontologies available in standard repositories proves the effectiveness of inductive classification using the kernel function integrated with a SVM.

The paper is organized as follows. After recalling the basics of the DLs representation (Sect. 2), we introduce relational kernels for the $\mathcal{ALCN}$ logic in Sect. 3. The application of kernel-based classification for inductive resource retrieval is presented in Sect. 4 and an experimental evaluation of the method is reported in Sect. 5. Finally, Sect. 6 concludes and outlines further applications and extensions of this work.

2 Reference Representation

The basics of the $\mathcal{ALCN}$ logic will be recalled (see [1] for a thorough reference). Such a logic is endowed with the basic constructors employed by the standard ontology languages adopted in the SW (such as OWL).

2.1 Knowledge Bases in Description Logics

Concept descriptions are inductively defined starting with a set $N_C = \{C, D, \ldots\}$ of *primitive concept* names, a set $N_R = \{R, Q, \ldots\}$ of *primitive roles* and a set of names for the *individuals* (objects, resources) $N_I = \{a, b, \ldots\}$. Complex descriptions are built using primitive concepts and roles and the language constructors. The set-theoretic semantics of these descriptions is defined by an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set, the *domain* of the interpretation, and $\cdot^{\mathcal{I}}$ is the *interpretation function* that maps each $A \in N_C$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each $R \in N_R$ to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Complex descriptions can be built in $\mathcal{ALCN}$ using the language constructors listed in Table 1, along with their semantics derived from the interpretation of atomic concepts and roles [1]. Note that the *open-world assumption* (OWA) is made.

Table 1. Syntax and semantics of concepts in the $\mathcal{ALCN}$ logic.

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
full negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
c. conjunction	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
c. disjunction	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
existential r.	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}((x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$
universal r.	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}((x,y) \in R^{\mathcal{I}} \to y \in C^{\mathcal{I}})\}$
at least r.	$\geq nR$	$\{x \in \Delta^{\mathcal{I}} \mid \lvert\{y \in \Delta^{\mathcal{I}} : (x,y) \in R^{\mathcal{I}}\rvert\} \geq n\}$
at most r.	$\leq nR$	$\{x \in \Delta^{\mathcal{I}} \mid \lvert\{y \in \Delta^{\mathcal{I}} : (x,y) \in R^{\mathcal{I}}\rvert\} \leq n\}$

A *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ contains a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. $\mathcal{T}$ is the set of definitions[1] $C \equiv D$, meaning $C^{\mathcal{I}} = D^{\mathcal{I}}$, where C is the concept name and D is its description. $\mathcal{A}$ contains assertions on the world state concerning the individuals, e.g. $C(a)$ and $R(a, b)$, meaning that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.

Example 2.1 (Royal Family). This example shows a knowledge base modeling concepts and roles related to the British royal family[2]:

$\mathcal{T} = \{$ Male $\equiv \neg$Female,
Woman $\equiv$ Human $\sqcap$ Female,
Man $\equiv$ Human $\sqcap$ Male,
Mother $\equiv$ Woman $\sqcap$ $\exists$hasChild.$\neg$Human,
Father $\equiv$ Man $\sqcap$ $\exists$hasChild.$\neg$Human,
Parent $\equiv$ Father $\sqcup$ Mother,
Grandmother $\equiv$ Woman $\sqcap$ $\exists$hasChild.$\neg$Parent,
Mother-w/o-daughter $\equiv$ Mother $\sqcap$ $\forall$hasChild.$\neg$Female,
Super-mother $\equiv$ Mother $\sqcap$ $\geq$ 3.hasChild $\}$

$\mathcal{A} = \{$ Woman(elisabeth), Woman(diana), Man(charles), Man(edward),
Man(andrew), Mother-w/o-daughter(diana),
hasChild(elisabeth, charles), hasChild(elisabeth,edward),
hasChild(elisabeth, andrew), hasChild(diana, william),
hasChild(charles, william) $\}$

2.2 Inference Services

Many inference services are supported by a growing number of DL resoners. The principal inference service amounts to assessing whether a concept *subsumes* another concept according to their semantics:

[1] More general definitions of concepts by means of inclusion axioms ($C \sqsubseteq D$) may also be considered.

[2] From Franconi's DLs course: `http://www.inf.unibz.it/~franconi/dl/course`

Definition 2.1 (subsumption). *Given two descriptions C and D, C subsumes D, denoted by $C \sqsupseteq D$, iff for every interpretation $\mathcal{I}$ it holds that $C^{\mathcal{I}} \supseteq D^{\mathcal{I}}$. When $C \sqsupseteq D$ and $D \sqsupseteq C$ then they are equivalent, denoted with $C \equiv D$.*

Normally subsumption is computed w.r.t. the interpretations satisfying the knowledge base. More expressive languages allo for the construction or role-hierarchies based on subsumption.

Another important inference, since we aim at inductive methods that manipulate single resources, is *instance checking*, that amounts to deciding whether an individual belongs to the extension of a given concept [1]. Another related inference is *retrieval* which consists in finding the extension of a given concept:

Definition 2.2 (retrieval). *Given an knowledge base $\mathcal{K}$ and a concept C, find all individuals a such that $\mathcal{K} \models C(a)$.*

Conversely, it may be necessary to find the concepts which an individual belongs to (*realization problem*), especially the most specific one:

Definition 2.3 (most specific concept). *Given an ABox $\mathcal{A}$ and an individual a, the* most specific concept *of a w.r.t. $\mathcal{A}$ is the concept C, denoted $\mathsf{MSC}_{\mathcal{A}}(a)$, such that $\mathcal{A} \models C(a)$ and for any other concept D such that $\mathcal{A} \models D(a)$, it holds that $C \sqsubseteq D$.*

For some languages, the MSC may not be expressed by a finite description [1], yet it may be approximated by a more general concept. Generally approximations up to a certain depth k of nested levels are considered, denoted MSC^k. A maximal depth approximation will be generically indicated with MSC^*.

2.3 Normal Form

Many semantically equivalent (yet syntactically different) descriptions can be given for the same concept. Equivalent concepts can be reduced to a normal form by means of rewriting rules that preserve their equivalence [1]. We will adopt a normal form derived from [3].

Some notation is necessary for naming the various parts of a description:

- $\mathsf{prim}(C)$ is the set of all the primitive concepts (or their negations) at the top-level of C;
- $\mathsf{val}_R(C) = C_1 \sqcap \cdots \sqcap C_n$ if there exists a value restriction $\forall R.(C_1 \sqcap \cdots \sqcap C_n)$ at the top-level of C, otherwise $\mathsf{val}_R(C) = \top$;
- $\mathsf{ex}_R(C)$ is the set of the descriptions C' appearing in existential restrictions $\exists R.C'$ at the top-level conjunction of C.
- $\mathsf{min}_R(C) = \max\{n \in \mathbb{N} \mid C \sqsubseteq (\geq n.R)\}$ (always a finite number);
- $\mathsf{max}_R(C) = \min\{n \in \mathbb{N} \mid C \sqsubseteq (\leq n.R)\}$ (if unlimited then $\mathsf{max}_R(C) = \infty$).

A normal form may be recursively defined as follows:

Definition 2.4 ($\mathcal{ALCN}$ normal form). *A concept description C is in $\mathcal{ALCN}$* normal form *iff $C = \bot$ or $C = \top$ or if $C = C_1 \sqcup \cdots \sqcup C_n$ with*

$$C_i = \bigsqcap_{P \in \mathsf{prim}(C_i)} P \sqcap \bigsqcap_{R \in N_R} \left[\forall R.\mathsf{val}_R(C_i) \sqcap \bigsqcap_{E \in \mathsf{ex}_R(C_i)} \exists R.E \sqcap \geq m_i^R.R \sqcap \leq M_i^R.R \right]$$

where, for all $i = 1, \ldots, n$, $m_i^R = \mathsf{min}_R(C_i)$, $M_i^R = \mathsf{max}_R(C_i)$, $C_i \not\equiv \bot$ and, for all $R \in N_R$, $\mathsf{val}_R(C_i)$ and every sub-description in $\mathsf{ex}_R(C_i)$ are, in their turn, in $\mathcal{ALCN}$ normal form.

This normal form can be obtained by means of a repeated application of equivalence preserving operations, namely replacing defined concepts with their definition as in the TBox and pushing the negation in the nested level (*negation normal form*).

Example 2.2 (normal form). The concept description
$C \equiv (\neg A_1 \sqcap A_2) \sqcup (\exists R_1.B_1 \sqcap \forall R_2.(\exists R_3.(\neg A_3 \sqcap B_2)))$
is in normal form, whereas the following is not:
$D \equiv A_1 \sqcup B_2 \sqcap \neg(A_3 \sqcap \exists R_3.B_2) \sqcup \forall R_2.B_3 \sqcap \forall R_2.(A_1 \sqcap B_3)$
where A_i's and B_j's are primitive concept names and the R_k's are role names.

3 Defining Kernels for $\mathcal{ALCN}$

A family of valid kernels for the space $\mathcal{X}$ of $\mathcal{ALCN}$ descriptions can be proposed, based on the family defined for $\mathcal{ALC}$ [9]. The definition is based on the AND-OR tree structure of the descriptions in normal form, like for the standard tree kernels [10] where similarity between trees depends on the number of similar subtrees (or paths unraveled from such trees). Yet this would end in a merely syntactic measure which does not fully capture the semantic nature of expressive DLs languages such as $\mathcal{ALCN}$.

Normal form descriptions can be decomposed level-wise into sub-descriptions. There are three possibilities for each level: the upper level is dominated by the disjunction of concepts that, in turn, are made up of a conjunction of complex or primitive concepts. In the following the definition of the $\mathcal{ALCN}$ kernels (parametrized on the decaying factor λ) is reported.

Definition 3.1 ($\mathcal{ALCN}$ kernels). *Given an interpretation $\mathcal{I}$ of $\mathcal{K}$, the $\mathcal{ALCN}$* kernel *based on $\mathcal{I}$ is the function $k_{\mathcal{I}} : \mathcal{X} \times \mathcal{X} \mapsto \mathbf{R}$ structurally defined as follows: given two disjunctive descriptions $D_1 = \bigsqcup_{i=1}^n C_i^1$ and $D_2 = \bigsqcup_{j=1}^m C_j^2$ in $\mathcal{ALCN}$ normal form:*
disjunctive descriptions:

$$k_{\mathcal{I}}(D_1, D_2) = \lambda \sum_{i=1}^{n} \sum_{j=1}^{m} k_{\mathcal{I}}(C_i^1, C_j^2)$$

with $\lambda \in]0, 1]$

conjunctive descriptions:

$$k_{\mathcal{I}}(C^1, C^2) = \prod_{\substack{P_1 \in \mathsf{prim}(C^1) \\ P_2 \in \mathsf{prim}(C^2)}} k_{\mathcal{I}}(P_1, P_2) \cdot \prod_{R \in N_R} k_{\mathcal{I}}((m^R_{C^1}, M^R_{C^1}), (m^R_{C^2}, M^R_{C^2})) \cdot$$
$$\prod_{R \in N_R} k_{\mathcal{I}}(\mathsf{val}_R(C^1), \mathsf{val}_R(C^2)) \cdot \prod_{R \in N_R} \sum_{\substack{C^1_i \in \mathsf{ex}_R(C^1) \\ C^2_j \in \mathsf{ex}_R(C^2)}} k_{\mathcal{I}}(C^1_i, C^2_j)$$

where $m^R_{C^i} = \mathsf{min}_R(C^i)$ *and* $M^R_{C^i} = \mathsf{max}_R(C^i)$, $i = 1, 2$.

numeric restrictions:

$$k_{\mathcal{I}}((m_C, M_C), (m_D, M_D)) = \frac{\min(M_C, M_D) - \max(m_C, m_D) + 1}{\max(M_C, M_D) - \min(m_C, m_D) + 1}$$

if $\min(M_C, M_D) > \max(m_C, m_D)$ *and* $k_{\mathcal{I}}((m_C, M_C), (m_D, M_D)) = 0$ *otherwise.*

primitive concepts:

$$k_{\mathcal{I}}(P_1, P_2) = k_{\mathrm{set}}(P_1^{\mathcal{I}}, P_2^{\mathcal{I}}) = |P_1^{\mathcal{I}} \cap P_2^{\mathcal{I}}|$$

where k_{set} *is the kernel for set structures defined in [10]. This case includes also the negation of primitive concepts using:* $(\neg P)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus P^{\mathcal{I}}$

This kernel computes the similarity between disjunctive as the sum of the cross-similarities between any couple of disjuncts from either description. The rationale for this kernel is that similarity between disjunctive descriptions is treated by taking the sum of the cross-similarities between any couple of disjuncts from either description. The term λ is employed to downweight the similarity of the sub-descriptions on the grounds of the level where they occur.

Following the normal form, the conjunctive kernel computes the similarity between two input descriptions distinguishing for factors corresponding to primitive concepts, universal, existential and numeric restrictions, respectively. These values are multiplied reflecting the fact that all the restrictions have to be satisfied at a conjunctive level. Not that the values computed for value and existential restrictions involve recursive calls to the kernel functions on less complex structures.

The similarity of the numeric restrictions is simply computed as a measure of the overlap between the two intervals. Namely it is the ratio of the amounts of individuals in the overlapping interval and those the larger one, whose extremes are minimum and maximum. Note that some intervals may be unlimited above: max $= \infty$. In this case we may approximate with an upper limit N greater than $|\Delta^{\mathcal{I}}| + 1$.

Finally, the similarity between primitive concepts is measured in terms of the intersection of their extension. Making the *unique names assumption* on the names of the individual occurring in the ABox $\mathcal{A}$, one can consider the *canonical* *interpretation* [1] $\mathcal{I}$, using $\mathsf{Ind}(\mathcal{A})$ as its domain ($\Delta^{\mathcal{I}} := \mathsf{Ind}(\mathcal{A})$). Therefore, the kernel can be specialized as follows: since the kernel for primitive concepts is essentially a set kernel we can set the constant λ_p to $1/\Delta^{\mathcal{I}}$ so that the cardinality of he intersection is weighted by the number of individuals occurring in the overall ABox. Alternatively, another choice could be $\lambda_P = 1/|P_1^{\mathcal{I}} \cup P_2^{\mathcal{I}}|$ which would weight the rate of similarity (the extension intersection) measured by the kernel with the size of the concepts measured in terms of the individuals belonging to their extensions.

Being partially based on the concept structure and only ultimately on the extensions of the concepts at the leaves, it may be objected that this kernel function can roughly grasp the concept similarity based on their semantics. This may be well revealed by the case of input concepts that are semantically almost equivalent yet structurally different. However, it must be pointed out that the process of rewriting for putting the concepts in normal form tends to eliminate these differences. More importantly, the ultimate goal for defining a kernel will be comparing individuals rather than concepts. This will be performed recurring to the most specific concepts of the individuals w.r.t. the same ABox. Hence, it was observed that semantically similar individuals will tend to share the same structures as elicited from the same source.

3.1 Discussion

The validity of a kernel depends on the fact that the function is *positive definite*. Positive definiteness can be also proved exploiting some closure properties of the class of positive definite kernel functions [11]. Namely, multiplying a kernel by a constant, adding or multiplying two kernels yields another valid kernel. We can demonstrate that the function introduced above is indeed a valid kernel for our space of hypotheses. Observe that the core function is the one on primitive concept extensions. It is essentially a set kernel [10]. The versions for top-level conjunctive and disjunctive descriptions are also positive definite being essentially based on the primitive kernel. Descending through the levels there is an interleaving of the employment of these function up the the basic case of the function for primitive descriptions.

Exploiting these closure properties it could be pr oven[3] that:

Proposition 3.1. *Given an interpretation $\mathcal{I}$, the function $k_{\mathcal{I}}$ is a valid kernel for the space $\mathcal{X}$ of $\mathcal{ALCN}$ descriptions in normal form.*

As regards efficiency, it is possible to show that the kernel function can be computed in time $O(|N_1||N_2|)$ where $|N_i|$, $i = 1, 2$, is the number of nodes of the concept AND-OR trees. It can computed by means of dynamic programming. Knowledge Base Management Systems, especially those dedicated to storing instances, generally maintain information regarding concepts and instances which may further speed-up the computation.

[3] Proof omitted for brevity.

The kernel can be extended to the case of individuals $a, b \in \mathsf{Ind}(\mathcal{A})$ simply by taking into account the approximations of their MSCs:

$$k_{\mathcal{I}}(a, b) = k_{\mathcal{I}}(\mathsf{MSC}^*(a), \mathsf{MSC}^*(b))$$

In this way, we move from a graph representation like the ABox portion containing an individual to an intensional tree-structured representation.

Observe that the kernel function could be specialized to take into account the similarity between different relationships. This would amount to considering each couple of existential and value restriction with one element from each description (or equivalently from each related AND-OR tree) and the computing the convolution of the sub-descriptions in the restriction. As previous suggested for λ, this should be weighted by a measure of similarity between the roles measured on the grounds of the available semantics. We propose therefore the following weight: given two roles $R, S \in N_R$: $\lambda_{RS} = |R^{\mathcal{I}} \cap S^{\mathcal{I}}|/|\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}|$.

As suggested before, the intersection could be measured on the grounds of the relative role extensions with respect to the whole domain of individuals, as follows: $\lambda_{RS} = |R^{\mathcal{I}} \cap S^{\mathcal{I}}|/|R^{\mathcal{I}} \cup S^{\mathcal{I}}|$. It is also worthwhile to recall that some DLs knowledge bases support also the so called *R-box* [1] with assertions concerning the roles, thus we might know beforehand that for instance $R \sqsubseteq S$ and compute heir similarity consequently.

The extension of the kernel function to more expressive DL is not trivial. DLs allowing normal form concept definitions can only be considered. Moreover, for each constructor not included in the $\mathcal{ALCN}$ logic, a kernel definition has to be provided.

Related distance measures can also be derived from kernel functions which essentially encode a notion of similarity between concepts and between individuals. This can enable the definition of various distance-based methods for these complex representations spanning from clustering to instance-based methods.

4 Inductive Classification and Retrieval

In this paper, a kernel method is used to solve the following classification problem:

Definition 4.1 (classification problem). *Given a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, the set of individuals* Ind *and a set of concepts $DC = \{C_1, \ldots, C_s\}$ defined on the grounds of those in $\mathcal{K}$, the primal problem to solve is: considered an individual $a \in \mathsf{Ind}$ determine the subset of concepts $\{C_1, \ldots, C_t\} \subseteq DC$ to which a belongs to.*

This classification problem can be also be regarded as a retrieval problem with the following dual definition:

Definition 4.2 (retrieval problem). *Given a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a query concept Q defined on the grounds of those in $\mathcal{K}$ and the set of individuals in the ABox $\mathsf{Ind}(\mathcal{A})$, the dual problem to solve is: find all $b \in \mathsf{Ind}(\mathcal{A})$ such that $\mathcal{K} \models Q(b)$.*

In the general learning setting, the target classes are disjoint. This is not generally verified in the SW context, where an individual can be instance of more than one concept in the hierarchy. To solve this problem, a new answering procedure is proposed. It is based on the decomposition of the multi-class problem into smaller binary classification problems (one per class). Therefore, a simple binary value set ($V = \{-1, +1\}$) can be employed, where ($+1$) indicates that an example x_i occurs in the ABox w.r.t. the considered concept C_j (namely $C_j(x_i) \in \mathcal{A}$); (-1) indicates the absence of the assertion in the ABox. As an alternative, it can be considered $+1$ when $C_j(x_i)$ can be inferred from the knowledge base, and -1 otherwise.

Another issue has to be considered. In the general classification setting an implicit assumption of *Closed World* is made. On the contrary, in the SW context the *Open World Assumption* is generally made. To deal with the OWA, the absence of information on whether a certain instance x_i belongs to the extension of concept C_j should not be interpreted negatively, as seen before, rather, it should count as neutral information. Thus, another value set has to be considered, namely $V = \{+1, -1, 0\}$, where the three values denote, respectively, assertion occurrence ($C_j(x_i) \in \mathcal{A}$), occurrence of the opposite assertion ($\neg C_j(x) \in \mathcal{A}$) and assertion absence in $\mathcal{A}$.

Occurrences can be easily computed with a lookup in the ABox. Moreover, as in the previous case, a more complex procedure may be devised by substituting the notion of occurrence (absence) of assertions in (from) the ABox with the one of derivability from the whole KB, i.e. $\mathcal{K} \vdash C_j(x_i)$, $\mathcal{K} \not\vdash C_j(x_i)$, $\mathcal{K} \not\vdash C_j(x_i)$ and $\mathcal{K} \not\vdash \neg C_j(x_i)$, respectively.

Although this may improve the precision of inductive reasoning, it is also more computationally expensive, since the simple lookup in the ABox must be replaced with instance checking. Hence, considered the query instance x_q, for every concept $C_j \in C$ the classifier will return $+1$ if x_q is an instance of C_j, -1 if x_q is an instance of $\neg C_j$, and 0 otherwise. The classification is performed on the ground of a set of training examples from which such information can be derived.

These results can be used to improve concept retrieval service. By classifying the individuals in the Abox w.r.t. all concepts, concept retrieval is performed exploiting an inductive approach. As will be experimentally shown in the following, the classifier, besides of having a comparable behavior w.r.t. a standard reasoner, is also able to induce new knowledge that is not logically derivable. Moreover it can be employed for the query answering task by determining, as illustrated above, the extension of a new query concept built from concepts and roles in the considered ontology.

Note that instance-checking, as performed by a reasoner is P-SPACE complete for the reference DL language [1]. Conversely, the inductive classification procedure is very efficient once the SVM has been trained. Most of the training time is actually devoted to the construction of the kernel matrix which gains a lot of speed-up exploiting statistics on concept extensions normally maintained by knowledge base management systems [12]. Moreover ad hoc data structures can be employed to make this process even more efficient.

5 Experimental Evaluation

The $\mathcal{ALCN}$ kernel function has implemented in Java and integrated with a support vector machine from the LIBSVM library[4].

In order to assess the value of the kernel integrated in a SVM at solving the retrieval problem, experiments have been carried out on a number of simple and more complex ontologies.

It is difficult to find in the literature similar methods for a comparison of the outcomes. A recent approach using simple kernels with SVM^{light} has been qualitatively evaluated [2]; unfortunately the data (drawn from two ontologies) are not publicly available. Namely the authors artificially populated a knowledge base and then assess the quality of the induced models for a selected number of concepts.

We preferred to carry out more extensive experiments on available knowledge bases with no random population involving all concepts and individuals of the ontology. As such experiments are more easily repeatable. The details of experimental settings and the outcomes are reported in the following.

5.1 Experimental Setup

The experiments have been performed on nine different ontologies (w.r.t. the domain and size) represented in OWL.

Namely, the FAMILY and UNIVERSITY ontologies were developed in our lab[5] and populated with real data; the FSM, SURFACE-WATER-MODEL, NEWTESTAMENTNAMES, SCIENCE, PEOPLE, NEWSPAPER and WINES ontologies were selected from the Protégé library[6]. Details about such ontologies are reported in Table 2. The number of individuals spans from 50 to 1000. However ontologies are normally measured in terms of triples; in the experiment we have ontologies whose size goes from hundreds up to ten thousands of triples.

These ontologies are represented in languages that are generally more complex than $\mathcal{ALCN}$. Hence, in order to apply the kernel function, constructs that are not allowed by $\mathcal{ALCN}$ were discarded during the computation of the MSCs of the individuals.

The inductive classification method was applied to each ontology; namely, the individuals were checked to assess if they were instances of the concepts in the ontology using the classifier induced by the SVM adopting the $\mathcal{ALCN}$ kernel function (initially λ was set to 1). The performance of the classifier was

[4] http://www.csie.ntu.edu.tw/~cjlin/libsvm

[5] http://lacam.di.uniba.it:8000/~nico/research/ontologymining.html

[6] http://protege.stanford.edu

Table 2. Ontologies employed in the experiments.

Ontology	DL lang.	#concepts	#object prop.	#datatype prop.
People	$\mathcal{ALCHIN}(D)$	60	14	1
University	$\mathcal{ALC}$	13	4	0
FSM	$\mathcal{SOF}(D)$	20	10	7
Family	$\mathcal{ALCF}$	14	5	0
Newspaper	$\mathcal{ALCF}(D)$	29	28	25
Wines	$\mathcal{ALCIO}(D)$	112	9	10
Science	$\mathcal{ALCIF}(D)$	74	70	40
S.-W.-M.	$\mathcal{ALCOF}(D)$	19	9	1
NTN	$\mathcal{SHIF}(D)$	47	27	8

evaluated comparing its responses to those returned by a standard reasoner[7] used as baseline.

Specifically, for each individual in the ontology the MSC was computed and enlisted in the set of training or test examples (individuals). For each concept, a model was built training the SVM with the kernel on the proper set of examples. Each test example was then classified applying the induced classifier. The experiment has been repeated for each concept adopting a ten-fold cross-validation procedure. Actually there were two sessions: in the first we tested the system on primitive and defined concepts in the ontology while in the second more complex random concepts were randomly built on the grounds of these concepts for testing the system.

For the evaluation, initially the standard measures of precision, recall, F-measure were considered. Yet in a setting complying with an open-world semantics cases when the resulting answer was unknown had to be considered more carefully, since they still might possibly imply a classification of an instance as relevant. Then we decided to measure a sort of alignment between the response given by the deductive reasoner and the one returned by our inductive classifier.

The running time (on a Core2Duo 2Ghz Linux box with 2GB RAM) goes from minutes to 1.2 hours for a run of 10-fold cross-validation procedure on the individuals belonging to a single ontology w.r.t. each test concept. That includes the time elapsed for getting the correct classification from the reasoner for the comparison.

Hence, for each concept in the ontology, the following parameters have been measured for the evaluation [7]:

- *match rate*: cases of individuals that got the same classification by both classifiers;
- *induction rate*: individuals that the classifier found to belong to the target concept or its negation, while this was not logically deducible;
- *omission error rate*: cases of individuals for which the inductive classifier omitted to determine whether they were instances (or not) of the target concept while this could be logically ascertained by the reasoner;

[7] Pellet ver. 1.5.1: `http://pellet.owldl.com`

Table 3. Outcomes of the concept classification experiments ($\lambda = 1$): average percentages and standard deviations.

Ontology	match rate	induction rate	om. error rate	com. error rate
People	86.61 (± 10.37)	5.40 (± 12.44)	7.99 (± 6.44)	0.0 (± 0.0)
University	78.94 (± 9.78)	11.40 (± 8.65)	1.76 (± 6.09)	7.90 (± 7.26)
FSM	91.72 (± 9.32)	0.72 (± 2.79)	0.0 (± 0.0)	7.56 (± 9.53)
Family	61.95 (± 20.36)	3.15 (± 11.37)	34.89 (± 22.88)	0.0 (± 0.0)
NewsPaper	90.33 (± 8.29)	0.0 (± 0.0)	9.67 (± 8.29)	0.0 (± 0.0)
Wines	95.58 (± 7.85)	0.43 (± 3.44)	3.99 (± 7.30)	0.0 (± 0.0)
Science	94.20 (± 7.91)	0.72 (± 1.61)	5.08 (± 8.31)	0.0 (± 0.0)
S.-W.-M.	87.09 (± 15.83)	6.73 (± 15.96)	6.18 (± 9.14)	0.0 (± 0.0)
N.T.N.	92.52 (± 24.71)	2.58 (± 8.44)	0.15 (± 3.90)	4.75 (± 11.28)

- *commission error rate*: amount of individuals labeled as instances of a given concept, while they (logically) do not belong to that concept or vice-versa.

Further experimental sessions, reported in the following section, were performed by varying the parameter λ. Besides, another experiment has regarded testing the performance of the classifier on random query concepts generated by composition of (primitive and defined) concepts in the knowledge base.

5.2 Outcomes

Classification with Concepts in the Knowledge Base. Table 3 reports the outcomes of this first experiment. The average rates obtained over all the concepts in each ontology are reported, jointly with their range.

It is important to note that, for every ontology, the commission error was quite low. This means that the classifier did not make critical mistakes, i.e. cases when an individual is deemed as an instance of a concept while it really is an instance of another disjoint concept. Particularly, the commission error rate is not null in case of University and FSM ontologies and consequently also the match rate is the lowest. It is worthwhile to note that these ontologies have also a limited number of individuals. Specifically, the number of concepts is almost similar to the number of individuals, which represents a difficult situation in which there is not enough information for separating the feature space and then produce correct classifications. However, also in these conditions, the commission error was quite low, the matching rate is considerably high and the classifier was even able to induce new knowledge.

Interestingly, it was noticed that the match rate increased with the increase of the number of individuals in the considered ontology with a consequent strong decrease of the commission error rate that tends to 0 in for the most populated ontologies. For almost all ontologies the SVM classifier is also able to induce new knowledge, i.e. to assign an individual to a concept when this could not be decided by the deductive reasoner due to the open-world semantics.

Table 4. Outcomes of the concept classification experiments ($\lambda = .5$): average percentages and standard deviations.

Ontology	match rate	induction rate	om. error rate	com. error rate
People	86.61 (± 10.37)	5.4 (± 12.44)	7.99 (± 6.44)	0.0 (± 0.0)
University	71.06 (± 13.36)	11.40 (± 8.65)	4.38 (± 15.18)	13.16 (± 8.56)
FSM	91.72 (± 9.32)	0.72 (± 2.79)	0.0 (± 0.0)	7.56 (± 9.53)
Family	61.55 (± 20.47)	3.55 (± 12.81)	34.89 (± 22.88)	0.0 (± 0.0)
Newspaper	90.38 (± 8.15)	0.0 (± 0.0)	9.62 (± 8.15)	0.0 (± 0.0)
Wines	95.15 (± 8.81)	0.65 (± 5.19)	4.21 (± 7.50)	0.0 (± 0.0)
Science	87.68 (± 12.71)	6.52 (± 12.61)	5.80 (± 9.85)	0.0 (± 0.0)
S.-W.-M.	86.18 (± 17.86)	8.01 (± 16.36)	5.81 (± 7.74)	0.0 (± 0.0)
NTN	90.52 (± 25.10)	4.27 (± 10.03)	4.90 (± 11.73)	0.31 (± 5.22)

For some ontologies some rates exhibit high standard deviations. In a careful insight of such cases, we found that this was due to cases of concepts with very few positive (negative) examples. This problem is made harder by the particular DL setting that allows many individuals to have an unknown classification w.r.t. some concepts because of the OWA. This is particularly true for the ontologies Family and University that were intentionally built as *sparse* knowledge bases (lots of class-membership assertions for the various individuals cannot be logically derived from the knowledge base).

Besides a stable behavior was also observed as regards the omission error rate which is very often non-null, yet very limited. This is due to a high number of training examples classified as unknown w.r.t. a certain class. To decrease the tendency to a conservative behavior of the classifier, a threshold could be introduced for the consideration of the training examples labeled with 0 (”unknown” classification).

The experiment has been repeated by setting the parameter λ of the kernel function to smaller values. For example, the average results when $\lambda = 0.5$ are reported in Table 4 (we omit the other results for sake of brevity). From this table, where the average rates w.r.t. the various ontologies are reported, we can note that the classification results are comparable with those of the previous experiment. Again it is possible to note that halving of the λ value does not generally influence the classification results.

Particularly, for ontologies with the lowest numbers of individuals (e.g. University, FSM) the match rate sometimes also decreases w.r.t. the classification performed using $\lambda = 1$.

Random Query Concepts. Another experiment has been carried out, to test the kernel classifier as a means for performing inductive concept retrieval w.r.t. new query concepts built from the considered ontology. The method has been applied to solve a number of retrieval problems using $\lambda = 1$ for the kernel function. To this purpose, 15 queries were randomly generated by means of

Table 5. Outcomes of the experiments with random query concepts ($\lambda = 1$): average percentages and standard deviations.

Ontology	match rate	induction rate	om. error rate	com. error rate
People	88.56 (± 9.30)	4.05 (± 10.50)	7.4 (± 6.26)	0.0 (± 0.0)
University	71.99 (± 12.15)	15.98 (± 8.18)	0.94 (± 4.97)	11.10 (± 11.22)
FSM	87.80 (± 10.83)	0.86 (± 2.39)	0.0 (± 0.0)	11.34 (± 10.80)
Family	66.33 (± 16.79)	4.53 (± 10.93)	29.14 (± 20.87)	0.0 (± 0.0)
Newspaper	77.91 (± 10.06)	0.0 (± 0.0)	22.09 (± 10.06)	0.0 (± 0.0)
Wines	94.33 (± 9.12)	0.0 (± 0.0)	5.67 (± 9.12)	0.0 (± 0.0)
Science	97.86 (± 1.61)	0.51 (± 1.36)	1.63 (± 1.64)	0.0 (± 0.0)
S.-W.-M.	80.39 (± 16.26)	13.40 (± 18.93)	6.21 (± 7.28)	0.0 (± 0.0)
NTN	90.58 (± 25.23)	2.18 (± 11.0)	7.19 (± 14.36)	0.40 (± 7.39)

Table 6. Outcomes of the experiments with random query concepts ($\lambda = .5$): average percentages and standard deviations.

Ontology	match rate	induction rate	om. error rate	com. error rate
People	86.71 (± 8.97)	3.97 (± 10.75)	9.33 (± 6.49)	0.0 (± 0.0)
University	63.95 (± 18.56)	17.16 (± 10.08)	2.14 (± 11.14)	16.76 (± 13.06)
FSM	84.25 (± 12.40)	0.90 (± 3.05)	0.0 (± 0.0)	14.85 (± 12.84)
Family	66.38 (± 11.87)	7.86 (± 16.71)	28.77 (± 18.12)	0.0 (± 0.0)
Newspaper	81.81 (± 14.11)	0.92 (± 4.58)	17.28 (± 12.39)	0.0 (± 0.0)
Wines	89.46 (± 15.41)	5.20 (± 14.21)	1.95 (± 1.48)	0.0 (± 0.0)
Science	97.15 (± 0.87)	0.90 (± 1.80)	1.63 (± 1.64)	0.0 (± 0.0)
S.-W.-M.	84.86 (± 15.94)	8.6 (± 15.42)	6.88 (± 6.42)	0.0 (± 0.0)
NTN	89.11 (± 25.91)	5.17 (± 12.59)	5.35 (± 14.19)	0.37 (± 9.48)

conjunctions / disjunctions of (3 thru 8) primitive and/or defined concepts of each ontology.

As for the previous experiment, a ten fold cross-validation setting was applied with the same nine ontologies. The individuals have been assigned to each of the three classes and the classifier induced by the SVM has been used to decide on the membership to the query class of the test individuals. The outcomes are reported in Table 5, from which it is possible to observe that the behavior of the classifier generally remains unvaried w.r.t. the previous experiment whose outcomes are reported in Table 3. As in the other experiments, they were repeated for different values of λ. Table 6, reports the outcomes of such experiments for the case when $\lambda = .5$.

Summarizing, the $\mathcal{ALCN}$ kernel function can be effectively used, jointly with a SVM, to perform inductive concept retrieval, guaranteeing almost null commission error and, interestingly, the ability to induce new knowledge. The performance of the classifier increases with the increase of the number of individuals populating the considered ontology and the homogeneity of their spread across the concepts in the ontology.

These results are comparable to those obtained on an overlapping pool of datasets with a nearest neighbor classification method based on a semantic distance [7].

6 Conclusions and Future Work

We investigated multi-relational learning techniques in the SW peculiar context. Specifically, a kernel function has been defined for $\mathcal{ALCN}$ descriptions which was integrated with a SVM for inducing a statistical classifier working with this complex representation. The resulting classifier was tested on inductive retrieval and classification problems. Experimentally, it was shown that its performance is not only comparable to a standard deductive reasoner, but it is also able to induce new knowledge, which is not logically derivable. Particularly, an increase in prediction accuracy was observed when the instances are homogeneously spread.

The induced classifier can be exploited for predicting or suggesting missing information about individuals, thus completing large ontologies. Specifically, it can be used to semi-automatize the population of an ABox. Indeed, the new assertions can be suggested to the knowledge engineer that has only to validate their inclusion. This constitutes a new approach in the SW context, since the efficiency of the statistical and numerical approaches and the effectiveness of a symbolic representation have been combined.

The main weakness of the approach is on its scalability towards more complex DLs. While computing MSC approximations might be feasible, it may be more difficult focusing on a normal form when comparing descriptions. Indeed, as long as the expressiveness increases, the gap between syntactic structure semantics of the descriptions becomes more evident. As a next step, we can foresee the investigation of defining kernels for more expressive languages w.r.t. $\mathcal{ALCN}$, e.g. languages enriched with (qualified) number restrictions and inverse roles [1].

The derivation of distance measures from the kernel function may enable a series of further distance-based data mining techniques such as clustering and instance-based classification. Conversely, new kernel functions can be defined transforming newly proposed distance functions for these representations, which are not language dependent and allow the related data mining methods to better scale w.r.t. the number of individuals in the ABox.

References

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

[2] S. Bloehdorn and Y. Sure. Kernel methods for mining instance data in ontologies. In K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *6th International Semantic Web Conference, ISWC 2007*, LNCS, pages 58–71, Busan, Korea, 2007. Springer.

[3] S. Brandt, R. Küsters, and A.-Y. Turhan. Approximation and difference in description logics. In D. Fensel, F. Giunchiglia, D. McGuinness, and M.-A. Williams, editors, *Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning, KR02*, pages 203–214. Morgan Kaufmann, 2002.

[4] P. Buitelaar, P. Cimiano, and B. Magnini, editors. *Ontology Learning from Text: Methods, Evaluation And Applications.* IOS Press, 2005.

[5] W.W. Cohen and H. Hirsh. Learning the CLASSIC description logic. In P. Torasso, J. Doyle, and E. Sandewall, editors, *Proceedings of the 4th International Conference on the Principles of Knowledge Representation and Reasoning*, pages 121–133. Morgan Kaufmann, 1994.

[6] C.M. Cumby and D. Roth. On kernel methods for relational learning. In T. Fawcett and N.Mishra, editors, *Proceedings of the 20th International Conference on Machine Learning, ICML2003*, pages 107–114. AAAI Press, 2003.

[7] C. d'Amato, N. Fanizzi, and F. Esposito. Query answering and ontology population: An inductive approach. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *Proceedings of the 5th European Semantic Web Conference, ESWC2008*, volume 5021 of *LNCS*, pages 288–302. Springer, 2008.

[8] F. Esposito, N. Fanizzi, L. Iannone, I. Palmisano, and G. Semeraro. Knowledge-intensive induction of terminologies from metadata. In F. van Harmelen, S. McIlraith, and D. Plexousakis, editors, *ISWC2004, Proceedings of the 3rd International Semantic Web Conference*, volume 3298 of *LNCS*, pages 441–455. Springer, 2004.

[9] N. Fanizzi and C. d'Amato. A declarative kernel for $\mathcal{ALC}$ concept descriptions. In F. Esposito, Z. W. Raś, D. Malerba, and G. Semeraro, editors, *In Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems, ISMIS2006*, volume 4203 of *LNAI*, pages 322–331. Springer, 2006.

[10] T. Gärtner, J.W. Lloyd, and P.A. Flach. Kernels and distances for structured data. *Machine Learning*, 57(3):205–232, 2004.

[11] D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, Department of Computer Science, University of California – Santa Cruz, 1999.

[12] I. R. Horrocks, L. Li, D. Turi, and S. K. Bechhofer. The instance store: DL reasoning with large numbers of individuals. In V. Haarslev and R. Möller, editors, *Proceedings of the 2004 Description Logic Workshop, DL 2004*, volume 104 of *CEUR Workshop Proceedings*, pages 31–40. CEUR, 2004.

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

[14] J. Lehmann and P. Hitzler. A refinement operator based learning algorithm for the $\mathcal{ALC}$ description logic. In H. Blockeel, J. Ramon, J. Shavlik, and P. Tadepalli, editors, *Proceedings of the 17th International Conference on Inductive Logic Programming, ILP2007*, volume 4894 of *LNCS*. Springer, 2008.