Integrating New Refinement Operators in Terminological Decision Trees Learning

Giuseppe Rizzo[1], Nicola Fanizzi[1], Jens Lehmann[2], and Lorenz Bühmann[3]

[1] LACAM - Università degli Studi di Bari, Via Orabona 4, 70125, Bari, Italy
{giuseppe.rizzo1,nicola.fanizzi}@uniba.it
[2] Computer Science Institute, Univ. of Bonn, Römerstr. 164, 53117 Bonn, Germany
jens.lehmann@cs.uni-bonn.de
[3] AKSW- Univerität Leipzig, Augustusplatz 10, 04109,Leipzig, Germany
buehmann@informatik.uni-leipzig.de

Abstract. The problem of predicting the membership w.r.t. a target concept for individuals of Semantic Web knowledge bases can be cast as a concept learning problem, whose goal is to induce intensional definitions describing the available examples. However, the models obtained through the methods borrowed from *Inductive Logic Programming* e.g. Terminological Decision Trees, may be affected by two crucial aspects: the refinement operators for specializing the concept description to be learned and the heuristics employed for selecting the most promising solution (i.e. the concept description that describes better the examples). In this paper, we started to investigate the effectiveness of Terminological Decision Tree and its evidential version when a refinement operator available in DL-Learner and modified heuristics are employed. The evaluation showed an improvement in terms of the predictiveness.

1 Introduction

In the context of the Semantic Web, the effectiveness of the reasoning on the knowledge represented in ontological form through languages derived from Description Logics (DLs) [1] formalism is affected by the inherent incompleteness due to the Open World Assumption

In the last years, resorting to machine learning methods have shown promising results for tackling this problem, for instance, by inducing predictive models to assess the membership of an individual w.r.t. a given concept for supporting various tasks such as (approximate) query answering and ontology completion. Despite the large availability of inductive methods for solving the problem [2], in this work (and similarly to other previous ones [3–5]) we focused on methods borrowed from *Inductive Logic Programming* (ILP) for solving the concept learning problem. These methods produce intentional definitions that describe the available instances that can be used for classifying them and therefore offering a trade-off between comprehensibility and predictiveness. In these methods, the learning is usually considered as a search process where the best solution as possible (i.e. the most accurate description among the possible ones describing the instances) is obtained via refinement operators to specialize or generalize the promising concept description, i.e. for obtaining a new concept description which

subsumes or is subsumed by the given one. Such methods,e.g. DL-FOIL [6], typically resorts to a *separate-and-conquer* strategy that aims at covering the largest number of positive instances excluding the negative ones. More recently, DL-LEARNER [7] has become a state-of-the-art framework that provides the implementation of various learning algorithms such as CELOE [8] and EL TREE LEARNER (ELTL) [9].

However separate-and-conquer methods suffers of some drawbacks. For instance, such methods learns one concept description at once. In addition, separate-and-conquer approaches tend to consider partial solutions more times yielding inefficient solutions for the learning problem. Finally, these methods may fail to induce the description when the learning problem is hard. On the other hand, *divide-and-conquer* strategies have been exploited to overcome such problems. Among divide-and-conquer solutions, it is possible to mention decision tree models, which have been devised for solving learning problems, also in the context of multi-relational data representations and, in particular, for knowledge bases modeled with Description Logics formalism. Such extensions are called *Terminological Decision Trees* [3]. Also, further extensions, namely *Evidential Terminological Decision Trees*, are able to represent the uncertainty and to handle the presence of tests with uncertain result by resorting to the Dempster-Shafer Theory [4, 5, 10]. In order to improve the quality of the aforementioned models, there are two crucial aspects that should be investigated: the refinement operator adopted to generates the candidate concept descriptions to be installed as a new node and the heuristics for selecting the best description [11]. Specifically, for both Terminological Decision Trees and their evidential version, the refinement operator used in [3–5] may not generate candidates that discerns the positive instances from the negative ones, likely due to the nature of the employed operator which exploits randomly generated sub-concepts and roles of a knowledge base. As a consequence, the resulting specializations may be not definitely related to the target concept and a large number of both missing values and misclassification cases may be found in the test phase. This problem affects also the values of the heuristic employed for selecting the best concept description: the candidates concepts have similar values of either information gain (in the case of the terminological decision trees) or non-specificity measure (in the case of the evidential terminological decision trees [4]). Moving from this idea, we carried out a preliminary analysis concerning the effectiveness of tree models endowed with another refinement operator and additional measures integrated into the heuristic employed for inducing the models. Specifically, we used a refinement operator adopted by CELOE and implemented in DL-LEARNER and introduced a regularized versions of the heuristics used for the best concept selection which is based on the Jaccard similarity.

The rest of the paper is organized as follows: Sect. 2 recalls the notion of DL knowledge bases and the refinement operator; Sect. 3 gives some notions about the Terminological Decision Trees and Evidential Terminological Decision Trees and describes the procedure for inducing *Terminological Decision Trees* that includes both the novel refinement operator and a Jaccard-based regularization term in the heuristic exploited for selecting the best concept, Sect. 4 proposes an empirical evaluation in order to understand the effectiveness of the proposed changes in the Decision Tree learning algorithms. Finally, conclusions and further outlooks are reported.

2 Basics

In this section we recall the notions concerning Description Logics and we introduce the class-membership prediction and the concept learning problems. Finally, we briefly provide some notions about the Dempster-Shafer Theory that are used by the extension of terminological decision tree considered in the paper.

2.1 Description Logics and Knowledge Bases

Description Logics (DLs) [1] are a family of knowledge representation languages exploited to model a domain in terms of *concepts* and *roles*. Given a set of atomic concept names $N_C = \{A, B, \cdots\}$ and roles $N_R = \{R, S, \cdots\}$, more complex concept descriptions (usually denoted by the letters $C, D, \cdots$) regarding a set of objects, named *individuals*, can be built by using a set of operators (e.g. complement, conjunction and disjunction between concepts). The set of operators adopted to build the concept descriptions determines the expressiveness of the representation language. In DLs, the knowledge about the domain is intensionally modeled by using a set of inclusion (*subsumption*) axioms between the concepts such as $C \sqsubseteq D$ (C is subsumed by D). Also, the domain can be described by a set of facts concerning the individuals. Such facts are called *concept and role assertions* and they are usually denoted by $C(a)$ and $R(a, b)$. Therefore, a DL *knowledge base* is a couple $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ where $\mathcal{T}$ is the TBox containing the intensional knowledge and $\mathcal{A}$ is the Abox containing the assertions. We will denote the set of individuals occurring in $\mathcal{A}$ by $\mathsf{Ind}(\mathcal{A})$.

Similarly to other first-order logic-based formalisms, the semantics is defined for each concept / role / individual by interpreting them according to the *model-theoretic semantics*. Formally, an interpretation is a couple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ composed by a non-empty set of objects representing the *domain* of the interpretation $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$ that maps: 1) each individual a to an object $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$; 2) each concept C to a subset $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$; 3) each role R to a subset $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The semantics of a complex description, say C is defined by applying recursively the interpretation function to the concepts used to build C. According to the model-theoretic semantics, an interpretation $\mathcal{I}$ *satisfies* an axiom $C \sqsubseteq D$ when $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ and an assertion $C(a)$ (resp. $R(a, b)$) when $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$). $\mathcal{I}$ is a *model* for $\mathcal{K}$ when it satisfies each axiom/assertion α in $\mathcal{K}$ ($\mathcal{I} \models \alpha$). When the axiom α is satisfied w.r.t. these models, we write $\mathcal{K} \models \alpha$. Various reasoning services are available for making new inferences from $\mathcal{K}$, which may involve either the TBox or the ABox. Among them, we recall the *instance-checking* inference service that is crucial from an inductive point of view: given an individual a and a concept description C the goal is determine if $\mathcal{K} \models C(a)$. The *Open World Assumption* (OWA) that is usually made in this context, may affect the ability to prove the truth of either $\mathcal{K} \models C(a)$ or $\mathcal{K} \models \neg C(a)$, as there may be possible to find different interpretations that satisfy either cases.

In the sequel we will denote by $sh\downarrow$ for a concept A (a role R), the set of direct (asserted) sub-classes (resp. sub-roles) of the atomic concept A (resp. role R). Besides, a role R is *applicable* when $\exists A \in N_C$ where $domain(R) \sqsubseteq A$ and there is no A' such that $domain(R) \sqsubseteq A' \sqsubseteq A$. Finally, we denote as $ar(R)$ a concept as $A \in N_C$ where $range(R) \sqsubseteq A$ and there is no A' such that $range(R) \sqsubseteq A' \sqsubseteq A$.

2.2 Class-membership prediction and Concept Learning Problem

The task of assessing the membership of an individual w.r.t. a target concept through inductive methods aims at approximating a function from the available instances that allows to determine if a an individual is an instance of the concept or not. A possible formalization of the problem, as proposed in [5], is reported below:

Definition 1 (Class-membership prediction problem).

Given

- *a target concept* C*;*
- *a label set* $\mathcal{L} = \{-1, 0, +1\}$
- *an error threshold* ϵ
- *a training set* $\mathsf{Tr} \subseteq \mathsf{Ind}(\mathcal{A})$ *of examples for which the correct classification value of* $t_{\mathsf{C}}(\cdot) : \mathsf{Ind} \rightarrow \mathcal{L}$ *is known, partitioned into positive, negative and uncertain-membership instances:*
  - $\mathsf{Ps} = \{a \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{K} \models C(a),$ *i.e.* $t_{\mathsf{C}}(a) = +1\}$,
  - $\mathsf{Ns} = \{a \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{K} \models \neg C(a),$ *i.e.* $t_C(a) = -1\}$
  - $\mathsf{Us} = \mathsf{Tr} \setminus (\mathsf{Ps} \cup \mathsf{Ns})$*, i.e.* $\{a \in \mathsf{Ind}(\mathcal{A}) : t_{\mathsf{C}}(a) = 0\}$*;*

Build *a classifier* $h_{\mathsf{C}} : \mathsf{Ind}(\mathcal{A}) \rightarrow \{-1, 0, +1\}$ *for* C *such that*

$$\frac{1}{|\mathsf{Tr}|} \sum_{a, \in \mathsf{Tr}} \mathbf{1}[h_{\mathsf{C}}(a) = t_{\mathsf{C}}(a)] > 1 - \epsilon$$

where $\mathbf{1}[\cdot]$ *is the* indicator *function returning* 1 *if the argument is true and* 0 *otherwise.*

To this purpose, various methods can be used for approximating this function, e.g. *non-parametric models* [2]. As an alternative, intensional descriptions of the available examples can be produced. Learning such descriptions is usually known as *concept learning problem* [11]. The concept learning problem in the context of a knowledge base can be formalized as follows.

Definition 2 (Concept Learning in DLs).

Given

- *the knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$*,*
- *a target concept* C*,*
- *the training set* $\mathsf{Tr} = \mathsf{Ps} \cup \mathsf{Ns} \cup \mathsf{Us}$*,*

Find *a concept description* D *approximating* C*, such that:*

- $\forall a \in \mathsf{Ps} : \ \mathcal{K} \models D(a)$
- $\forall b \in \mathsf{Ns} : \ \mathcal{K} \models \neg D(b)$

Therefore the goal of learning process is to find a concept description that is correct w.r.t. the examples. One could not be interested to a solution that fit perfectly to the training individuals but to induce a description general enough for classifying new individuals. Concept learning can be regarded as a search process in the space of concepts $\mathcal{S}$, which can be explored by imposing a quasi-ordering between DL concepts, i.e. a

reflexive and transitive relation and then to use a *refinement operator* which maps a concept onto a set of other concepts. In the following, we consider the subsumption relation $\sqsubseteq$ between concepts as a quasi-ordering relation.

The definition of the refinement operator is reported below:

Definition 3. *Given a quasi-ordered space $(\mathcal{S}, \sqsubseteq)$, a downward (resp. upward) refinement operator ρ is mapping from S to $2^{\mathcal{S}}$ such that for any concept description $C \in \mathcal{S}$ and $C' \in \rho(C)$, $C' \sqsubseteq C$ (resp. $C \sqsubseteq C'$)*

2.3 The Dempster-Shafer Theory

One of the models exploited in this paper is a modified version of terminological decision trees endowed with the operators of the Dempster-Shafer Theory (DST) [10]. Therefore, for sake of completeness, we shortly recall the basic notions of this theory used by such predictive models.

The DST is regarded as a generalization of probability theory. In the DST, a domain is usually represented through a *frame of discernement*, denoted by Ω, i.e. a set of mutually and exhaustive hypotheses. For our purposes, the frame of discernment represents the set of admissible membership values w.r.t. the target concept C, i.e. $\Omega = \{-1, +1\}$.

Given the frame of the discernment, a *Basic Belief Assignment* (BBA) can be build, that is a mapping $m : 2^{\Omega} \rightarrow [0, 1]$ such that $\forall A \in 2^{\Omega}\ m(A) > 0$ if $A \neq \emptyset$ and $\sum_{A \in 2^{\Omega}} m(A) = 1$. The value of a BBA function for a set of hypotheses A conveys the amount of belief exactly assigned to A but not to its subsets. In the DST, knowing the BBA allows to determine the *belief* and the *plausibility functions*. The belief function is a mapping $Bel : 2^{\Omega} \rightarrow [0, 1]$ such that $\forall A \in 2^{\Omega}\ Bel(A) = \sum_{B \subseteq \mathcal{A}} m(B)$ represents the total amount of belief assigned to A given the available evidences. The plausibility function is a mapping $Pl : 2^{\Omega} \rightarrow [0, 1]$ such that $\forall A \in 2^{\Omega}\ Pl(A) = \sum_{B \cap \mathcal{A} \neq \emptyset} m(B)$ and it quantifies the total amount of belief in favor of a set of hypotheses A when further evidences are available.

Other important notions concern the *non-specificity measure* [12] and *the combination rules* [13]. Given a BBA m the non-specificity measure $Ns(m)$ quantifies the degree of imprecision about the knowledge about a set of hypotheses, i.e. $Ns(m) = \sum_{A \subseteq \Omega, A \neq \emptyset} m(A) \log |A|$. A large non specificity measure denotes high uncertainty and imprecision about the available knowledge. As regards the combination rules, they represent operators used to pool BBAs coming from heterogeneous sources of information. The literature proposed various approaches for combining BBAs [13]. Among them, the *Dubois-Prade combination rule* has been adopted in the evidence-based version of a terminological decision tree [14]. The operator pools two BBAs, m_1 and m_2 as follows: $\forall A \in 2^{\Omega}\ m_{12}(A) = \sum_{B \cup C = A} m_1(B) m_2(C)$.

3 Learning Tree models in DLs

3.1 The models

The class-membership prediction task can be tackled by inducing either *Terminological Decision Trees* (TDTs) [3] or *Evidential Terminological Decision Trees* (ETDTs) [4].

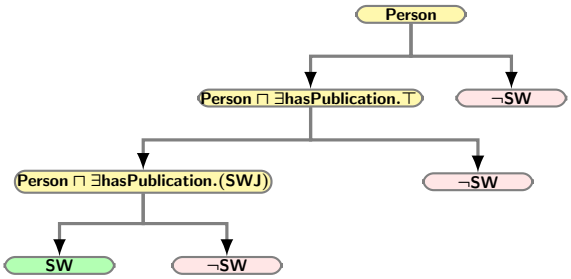

Fig. 1. A TDT for deciding if a person is a researcher that works in the field of the Semantic Web

Definition 4 (Terminological Decision Tree). *Given the knowledge base $\mathcal{K}$, a* Terminological Decision Tree *is a binary tree where:*

- *each intermediate node contains a conjunctive concept description D that stands for a test;*
- *each leaf contains a label used to denote the (positive/negative) membership w.r.t. the target concept C*
- *the branches correspond, respectively, to the result of the test performed over D (resp. $\neg D$);*

As illustrated in [3], a TDT can be used to learn concept descriptions and to determine the membership for an unseen individual. However, as argued in [4], when a TDT is used for predicting the class-membership for a new individual, the models cannot assign a definite membership due to intermediate tests with an unknown result. This is similar to the presence of missing values for decision trees targeting attribute-value datasets. In order to take into account this aspect, Evidential Terminological Decision Trees (ETDTs) have been devised [4, 5]. They are defined as an extension of the TDTs [3] based on the evidential reasoning [10].

Definition 5 (Evidential Terminological Decision Tree). *Given the knowledge base $\mathcal{K}$, an* Evidential Terminological Decision Tree *is a binary tree where:*

- *each intermediate node contains a pair (D, m) where D is a conjunctive concept description that stands for a test and m is used to describe the membership w.r.t. D;*
- *each leaf contains both the label and the BBA m used to describe the membership w.r.t C;*
- *the branches correspond, respectively, to the result of the test performed over D (resp. $\neg D$);*

Fig. 1 and Fig. 2 report two examples of a TDT and an ETDT that are used for deciding the membership of an individual w.r.t. the target concept *Semantic Web researcher* (SW). The models can be used for deciding if an individual is a researcher whose topic concerns the Semantic Web.

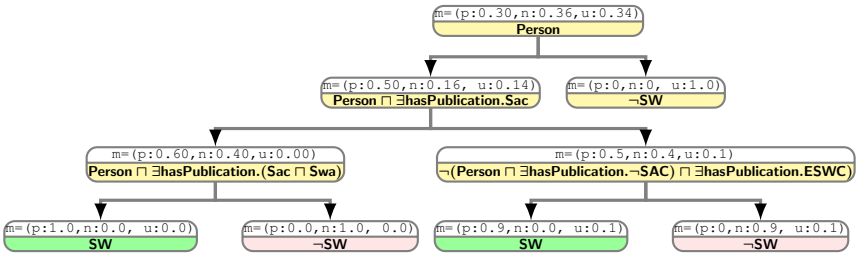

Fig. 2. An ETDT for deciding if a person is a Semantic Web researcher

3.2 Training

Given the concept C (used a label to be installed as a leaf) and the training set $\mathsf{Tr} = \langle \mathsf{Ps}, \mathsf{Ns}, \mathsf{Us} \rangle$, the methods for inducing both TDTs and ETDTs apply a *divide-and-conquer* strategy (see [3–5] for further details). The methods perform a recursive partitioning of the training set where, at each level, the individuals are grouped according to the results of some instance-check tests w.r.t. the most promising concepts description. The process is repeated until the instances sorted to a node have the same definite membership w.r.t. C. The concept descriptions that are installed as nodes during the training step are generated by specializing the concept installed as father node and passed as an input for the algorithm. Among the possible candidates, the algorithms select the best description according to a certain heuristic. In the case of ETDTs, the algorithm generates for the current node both a concept and a BBA estimating by using relative frequencies of the positive, negative and uncertain-membership instances routed to the node. Two examples of the learning procedures for TDTs and ETDTs are reported below.

Example 1 (Inducing TDTs). As regards the induction of the TDT reported in Fig. 1, the concept Person is installed as root note. The first refinement that is installed as a left-child node is $\mathsf{Person} \sqcap \exists \mathsf{hasPublication}.\top$, which describes all the instances of the concept Person with a publication. This concept description is obtained by adding an existential restriction as a conjunct. The concept $\mathsf{Person} \sqcap \exists \mathsf{hasPublication}.\top$ installed as new node is further specialized by using the instances with a positive membership w.r.t the concept, resulting in the concept $\mathsf{Person} \sqcap \exists \mathsf{hasPublication}.\mathsf{SWJ}$ where a new concept name is introduced, namely SWJ (the concept used to denote the papers appeared in the *Semantic Web Journal*). Again, this concept is installed as left-child node.

Example 2 (Inducing ETDTs). Also the induction of the ETDT reported in Fig 2 starts from the concept Person. In this case, the first refinement that is installed into the left-child node is $\mathsf{Person} \sqcap \exists \mathsf{hasPublication}.\mathsf{SAC}$, where SAC is a new concept name concerning all papers appeared in SAC proceedings. The instances reached the node are then split according to the instance-check test results, and the concept is further specialized so that the concept $\mathsf{Person} \sqcap \exists \mathsf{hasPublication}.\mathsf{SAC} \sqcap \mathsf{SWA}$ is obtained, where SWA is related to those papers presented in the *Semantic Web Application Track*. After the installing of the new node and the further split of the training instances, the next

node that is installed as a leaf. The other branches of the trees can be obtained likewise. In addition, we can observe that the BBA m assigned to each intermediate node has a decreasing level of non-specificity measure w.r.t. the previous level.

Refinement operators As introduced in Sect. 1, refinement operators play a fundamental role for determining the strategy to navigate the concepts space and, in the case of TDTs and ETDTs, for obtaining the candidates concepts to be chosen and installed into the nodes. The examples reported above induce the trees by using the downward refinement operator adopted in [3] and [4] that generate specializations in one of the following forms: 1) by introducing a new concept name (or its complement as conjunct); 2) by refining a sub-description in the scope of an existential restriction; 3) by refining a sub-description in the scope of an universal restriction. This naïve refinement operator exploits concept names and roles without considering information like the concept hierarchy asserted in a knowledge base. Conversely, the refinement operator implemented in DL-LEARNER framework (that contains the implementation of various ILP-based learning algorithms) [7] consider this aspect and can be also extended for addressing various DL expressiveness.

Fig. 3.2 describes the refinement operator employed in this work: M_B is the set of the specializations of $\top$ obtained without resorting to disjunction operator that are not disjoint from $B \in \{\top\} \cup N_C$. This means that M_B contains concept in one of the following forms:

- $A \in N_C$ where $A \sqcap B \neq \bot$ and $A \sqcap B \neq \bot$ and there is no $A' \in N_C$ such that $A' \sqsubseteq A$
- $\neg A \in N_C$ where $\neg A \sqcap B \neq \bot$ and $\neg A \sqcap B \neq \bot$ and there is no $A' \in N_C$ such that $A \sqsubseteq A'$
- $\forall R.\top$, where R is the most general applicable role for B , i.e. there is no applicable role R' such that $R \sqsubseteq R'$
- $\exists R.\top$, where R is the most general applicable role for B

The ρ operator generates the specializations as follows. Firstly, it delegates the refinement process to an operator $\rho_B(\cdot)$, Using the index B allows to exclude the concepts that are disjoint with B. At the beginning $B = \top$. The $\rho_\top(\cdot)$ distinguishes various cases: the simplest cases concern the generation of the refinements for $\bot$ and $\top$. For $\bot$, the specialization process ends by returning an empty set of concepts. In the case of the refinement of $\top$, the operator returns disjunction of concepts C_i where $C_i \in M_B(C)$. Additional cases concern the refinement of an atomic concept A or its negation. For the atomic concept, the refinement operator returns two sets of specializations: the first set contains sub-concepts A' such that $A' \sqsubseteq A$, i.e. $A' \in sh \downarrow (A)$, while the second set contains concepts obtained through the conjunction of the concept A and concepts $D \in \rho_B(\top)$. The case of the complement of an atomic concept is tackled dually to the previous one but the operator generates also refinements in the form $\neg A'$ where $A' \in sh \uparrow (A)$.

The third case concerns the refinement of a concept in the form of an existential restriction $C = \exists R.D$ [4]. The operator produces three kinds of refinements: the first

[4] The refinement operator was originally devised to consider $\mathcal{ALC}$ expressiveness

one is obtained by replacing the sub-description D with a sub-description E that is a concept subsumed by D and it is not disjoint with the range of the role R; the second kind of refinements is obtained by replacing the sub-description D with the one in the form $D \sqcap E$, where E is a refinement contained into the set of specializations of $\top$; the third kind of refinements are obtained by replacing the role R with a sub-role S, i.e. $S \in sh\downarrow(R)$.

The fourth case described in Fig.3.2 illustrates the case of a concept in the form of an universal restriction, i.e. $C = \forall R.D$. This case is substantially dual to the case of existential restriction except for the specializations in the form $\forall R.\bot$ generated for the atomic concepts that have no sub-concepts. The last two cases concern concepts in conjunctive and disjunctive forms. In the first case, the refinement operator generates specializations by replacing a sub-description C_i with its refinements obtained by applying recursively the refinement operators. In the second case, the refinement operator produces specializes not only the various concept sub-description C_i (as in the case of conjunctive concept descriptions) but also it adds a new concept D as a conjunct.

Example 3 illustrates a simple example about the generation of the specializations.

Example 3 (ρ refinements). Let the following knowledge base be given:

$$\begin{aligned}\mathcal{K} = \{&\mathsf{Man} \sqsubseteq \mathsf{Person}, \mathsf{Woman} \sqsubseteq \mathsf{Person}, \mathsf{ESWC} \sqsubseteq \mathsf{Publication}\\ &\mathsf{EKAW} \sqsubseteq \mathsf{Publication}, \mathsf{EKAW} \sqcap \mathsf{ESWC} \equiv \bot\\ &\mathsf{domain}(\mathsf{hasFirstAuthor}) = \mathsf{Publication},\\ &\mathsf{range}(\mathsf{hasFirstAuthor}) = \mathsf{Person} \qquad \}\end{aligned}$$

The refinement operator generates the following refinements for $\top$:

$$\begin{aligned}\rho(\top) = \quad \{&\mathsf{Person}, \mathsf{Publication}, \neg\mathsf{Man}, \neg\mathsf{Woman},\\ &\neg\mathsf{EKAW}, \neg\mathsf{ESWC},\\ &\forall \mathsf{hasFirstAuthor}.\top, \exists \mathsf{hasFirstAuthor}.\top, \dots \}\end{aligned}$$

By using ρ, it is possible to specialize the concept $\mathsf{Publication} \sqcap \exists \mathsf{hasFirstAuthor}.\mathsf{Person}$ generating the following set of concept descriptions:

$$\begin{aligned}\rho(\mathsf{Publication} \sqcap \exists \mathsf{hasFirstAuthor}.\mathsf{Person}) = \quad \{&\mathsf{Publication} \sqcap \exists \mathsf{hasFirstAuthor}.\mathsf{Man}\\ &\mathsf{Publication} \sqcap \exists \mathsf{hasFirstAuthor}.\mathsf{Woman}\\ &\mathsf{EKAW} \sqcap \exists \mathsf{hasFirstAuthor}.\mathsf{Person},\\ &\mathsf{ESWC} \sqcap \exists \mathsf{hasFirstAuthor}.\mathsf{Person}, \dots \quad \}\end{aligned}$$

Note that the number of the possible specializations that are generated at each step via the refinement operator is *infinite* [11]. To overcome the problem, various strategies can be employed, e.g. by limiting the length of the specializations[5].

[5] The length of of a concept C, len(C) can be defined inductively as:

- len(A)=len($\top$)=len($\bot$)=1
- len($\neg D$)= len(D)+1
- len($D \sqcap E$)= len($D \sqcup E$)= len(D)+ len(E)+1
- len($\exists R.D$)= len($\forall R.D$)+1

$$\rho(C) = \begin{cases} \{\top\} \cup \rho_\top(C) & \text{if } C = \top \\ \rho_\top(C) \end{cases}$$

$$\rho_\top(C) = \begin{cases} \emptyset & \text{if } C = \bot \\ \{C_1 \sqcup C_2 \sqcup \cdots C_n | C_i \in M_B(C)\} & \text{if } C = \top \\ \{A' | A' \in sh \downarrow (A)\} \\ \cup \{A \sqcap D | D \in \rho_B(\top)\} & \text{if } C = A (A \in N_C) \\ \\ \{\neg A' | A' \in sh \uparrow (A)\} \\ \cup \{\neg A \sqcap D | D \in \rho_B(\top)\} & \text{if } C = \neg A (A \in N_C) \\ \\ \{\exists R.E | E = ar_A(R), E \in \rho_A(D)\} \\ \cup \{\exists R.D \sqcap E | E \in \rho_B(\top)\} \\ \cup \{\exists R'.D | R' \in sh \downarrow (R)\} & \text{if } C = \exists R.D \\ \\ \{\forall R.E | E = ar_A(R), E \in \rho_A(D)\} \\ \cup \{\forall R.D \sqcap E | E \in \rho_B(\top)\} \\ \cup \{\forall R.\bot | D = A \in N_C \text{ and } sh \downarrow (A) = \emptyset\} \\ \cup \{\exists R'.D | R' \in sh \downarrow (R)\} & \text{if } C = \forall R.D \\ \\ \{C_1 \sqcap C_2 \sqcap \cdots \sqcap C_{i-1} \sqcap D \sqcap C_{i+1} \cdots C_n | \\ D \in \rho_B(C_i), 1 \leq i \leq n\} \\ & \text{if } C = C_1 \sqcap \cdots \sqcap C_n \\ \\ \{C_1 \sqcup C_2 \sqcup \cdots \sqcup C_{i-1} \sqcup D \sqcup C_{i+1} \cdots C_n | D \in \rho_B(\top), 1 \leq i \leq n\} \\ \cup \{(C_1 \sqcup C_2 \sqcup \cdots \sqcup C_{i-1} \sqcup C_i \sqcup C_{i+1} \cdots C_n) \sqcap D | \\ D \in \rho_B(\top), 1 \leq i \leq n\} \\ & \text{if } C = C_1 \sqcup \cdots \sqcup C_n \end{cases}$$

Fig. 3. The refinement operator available in DL-Learner. Image adapted from [11]

Heuristics for the best candidate selection The heuristics used for the concept selection aim at maximizing a purity criterion. This idea, borrowed from the algorithm for the induction of decision trees, is used by TDT. In fact, during the induction of TDTs *information gain* is the criterion used for selecting the best concept description [3]. Instead ETDTs exploits an heuristic based on the minimization of non-specificity measure in order to determine the concept sub-description with the most definite membership [4]. However, both the information gain and the non-specificity measure do not consider aspects such as the complexity of the concept description or the similarity w.r.t. the concept installed into the father node. In the latter case, adding a sub-description that is not similar to the one installed into the father node may increase the risk that most instances are sent along a branch, leading to an error-prone classification model, or that a large number of missing values may be found. To penalize these concept descriptions, we can adopt the idea proposed in [15]: introducing a regularization terms in the information gain/non-specificity measure value. This is basically a *discounting factor* for the purity-measure employed for selecting the concept. As regards the information gain, let C and D two concepts installed into a father and a child node, the regularized version of information gain can be computed as

$$Gain(C, D) = c\left(H(C, \mathsf{Tr}) - \frac{n^l}{n} H\left(D, \mathsf{Ps}^l \cup \mathsf{Ns}^l \cup \mathsf{Us}^l\right) - \frac{n^r}{n} H\left(D, \mathsf{Ps}^r \cup \mathsf{Ns}^r \cup \mathsf{Us}^r\right)\right) \quad (1)$$

where n^l (resp. n^l) is the number of training individuals sent to the left (resp. right) branch, H is the entropy of the concept adopted as a test computed over a set of indi-

viduals and $c \in [0, 1]$ represents the aforementioned regularization factor. In this paper, the regularization factor takes into account the similarity w.r.t. the concept installed into the father node and it is computed through the Jaccard similarity between the set of the individuals which belong to those concepts. $J(C, D) = \frac{|ret.(C) \cap ret.(D)|}{|ret.(C) \cup ret.(D)|}$ where $ret.(E)$ for a given concept E is the set of individuals which belongs to E. Similarly to the case of information gain, a regularized version of the non specificity measure can be defined.

3.3 Classification

In order to make prediction with the produced models, we consider a ternary classification problem for assessing the membership of an individual [3, 4]. The strategy is based on the navigation of tree structure according to the instance-check results. The algorithms start from the root and follows either the left or the positive branch according to the results of the instance check test w.r.t. the concept. The algorithms differ in the strategies exploited for coping with the case of uncertain results w.r.t. the intermediate tests: while the exploration of a TDT is stopped by assigning the uncertain-membership label for the test individual, both branches departing from the node with an uncertain result are navigated in order to reach more leaves when an ETDT is used to classify an individual. In this case, the algorithm collects the BBAs contained into the leaves that are subsequently pooled according to the Dubois-Prade rule [4].

Example 4 (Classification through TDTs). Given the TDT reported in Fig. 1 and a new individual a. Assuming that for this individual the membership w.r.t. the target concept SW is unknown but, according to the available knowledge base, it is an instance of the concept Person and a publication in SWJ exists, the classification algorithm will follow the most-left path of the tree and it will classify the individual as a positive instance. Conversely, $\mathsf{Person}(\mathsf{a})$ is entailed from the knowledge base but the it cannot determine if $\mathsf{Person} \sqcap \exists \mathsf{hasPublication}.\top(\mathsf{a})$ the classification algorithm stop the traversing of the tree assigning the uncertain-membership value.

Example 5 (Classification through ETDTs). The model proposed in Fig. 2 can be used for classifying an individual a that is an instance of the concept Person. The traversing process checks the membership w.r.t. the intermediate concept description.If neither of the encountered tests is satisfied and the individual is a instance of their complement concept, the algorithm follows the most-right path collecting the BBA of the single leaf and then, computing Bel function and assigning the class that corresponds to the hypothesis with the largest belief value. It is straightforward to note that the classification procedure will decide in favor of the negative membership. On the other hand, if an intermediate test with an uncertain result is encountered, e.g. it cannot be determined if a is an instance of either the concept $\mathsf{Person} \sqcap \mathsf{hasPublication}.\top$ or its complement. In this case, the algorithm explores both the left sub-tree, whose root contains the concept description $\mathsf{Person} \sqcap \mathsf{hasPublication.SAC}$, and the right branch, whose root contains the concept $\neg(\mathsf{Person} \sqcap \mathsf{hasPublication.SAC}) \sqcap \mathsf{hasPublication.ESWC}$. Following these branches, the algorithm can collect up to 4 BBAs (if there are further uncertain test results) that are combined according to the Dubois-Prade rule

Table 1. Ontologies employed in the experiments

Ontology	*Expressiv.*	*# Classes*	*# Roles*	*# Individ.*
Lymph	$\mathcal{AL}$	53	0	148
NTN	$\mathcal{SHIF}(\mathcal{D})$	47	27	676
MUTAGENESIS	$\mathcal{AL}(\mathcal{D})$	86	5	14145
CARCINOGENESIS	$\mathcal{ALC}(\mathcal{D})$	142	4	22372

4 Empirical Evaluation

In this section we report the settings and the outcomes of an empirical evaluation, where we compared TDTs and ETDTs w.r.t. to other methods implemented in DL-LEARNER [7].

4.1 Setup

In our experiments, we considered various Web ontologies, whose dimensions and expressiveness are reported in Tab.1. LYMPH represents an OWL porting of the Lymphography dataset, which is available at the UCI repository (`http://archive.ics.uci.edu/ml/`). Instead, NTN is an ontology concerning the characters of the New Testament. MUTAGENESIS and CARCINOGENESIS are the porting of the well known datasets typically employed to test ILP methods.

For each ontology, we considered the learning problems available with the DL-LEARNER release (`http://www.dllearner.org`). Specifically, for LYMPH, we considered the learning problems contained in lymphography_Class2. conf. Instead, for NTN the learning problem aims at discovering if the ethnicity of an individual is Jewish. Finally, for MUTAGENESIS and CARCINOGENESIS the tasks aim at predicting if a chemical compound is mutagenic and carcinogenic, respectively. In the evaluation, TDTs and ETDTs have been compared against CELOE and ELTL DISJUNCTIVE. For the induction of trees we tested the original models against the new versions endowed with further refinement operators and the Jaccard similarity as a regularization term. As regards the refinement operators, we resort to both the original operator employed in [3] and [4], the RHO operator available in DL-LEARNER with a maximum length of 2. We used a 10-fold cross validation for assessing the performance of the algorithms.

The performance has been compared in terms of F-measure and other metrics that take into account the Open World Assumption [3, 4], which are based on a comparison between inductive classification and the answer of a reasoner (PELLET: `http://clarkparsia.com/pellet`). The metrics are: 1) *match* (M), i.e. the rate of the test examples for which the inductive model and a reasoner predict the same membership (i.e. $+1$ vs. $+1$, -1 vs. -1, 0 vs. 0); 2) *commission*(C), i.e. the rate of the test examples for which predictions are opposite (i.e. $+1$ vs. -1, -1 vs. $+1$); 3) *omission* (O), i.e. the rate of test examples for which the inductive method cannot determine a definite membership (-1 or $+1$) while the reasoner is able to do it; 4) *induction* (I), i.e. rate of test examples where the inductive method can predict a definite membership while it is not logically derivable.

Table 2. Results of the experiments

Ontology	Index	TDT		ETDT		CELOE	ELTL
		original	regularized+ rho	original	regularized+ rho		
Lymph	F_1	18.00 ± 33.27	$\mathbf{100.00 \pm 00.00}$	63.56 ± 22.38	$\mathbf{70.76 \pm 01.55}$	87.18 ± 08.29	100.00 ± 00.00
	M%	17.00 ± 19.15	$\mathbf{54.73 \pm 01.87}$	53.52 ± 03.87	$\mathbf{54.76 \pm 01.87}$	52.00 ± 03.60	54.77 ± 01.87
	C%	00.00 ± 00.00	00.00 ± 00.00	46.48 ± 03.87	$\mathbf{45.23 \pm 01.87}$	12.91 ± 07.71	00.00 ± 00.00
	O%	83.00 ± 19.15	$\mathbf{45.23 \pm 01.87}$	00.00 ± 00.00	00.00 ± 00.00	35.08 ± 05.78	45.23 ± 01.87
	I%	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00
NTN	F_1	30.00 ± 48.31	$\mathbf{100.00 \pm 00.00}$	40.00 ± 51.64	$\mathbf{100.00 \pm 00.00}$	100.00 ± 00.00	37.95 ± 05.97
	M%	29.47 ± 47.48	$\mathbf{99.47 \pm 01.66}$	85.59 ± 12.96	$\mathbf{100.00 \pm 00.00}$	99.47 ± 01.66	22.85 ± 06.42
	C%	00.00 ± 00.00	00.00 ± 00.00	14.41 ± 12.96	$\mathbf{00.00 \pm 00.00}$	00.00 ± 00.00	69.27 ± 18.87
	O%	70.53 ± 47.48	$\mathbf{00.53 \pm 01.66}$	00.00 ± 00.00	00.00 ± 00.00	00.53 ± 01.66	07.90 ± 24.96
	I%	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00
MUTAGENESIS	F_1	00.00 ± 00.00	$\mathbf{70.43 \pm 00.02}$	70.43 ± 00.17	70.43 ± 00.17	94.00 ± 03.85	70.43 ± 00.17
	M%	00.00 ± 00.00	$\mathbf{54.36 \pm 00.20}$	54.36 ± 00.20	54.36 ± 00.20	93.03 ± 04.53	54.36 ± 00.20
	C%	00.00 ± 00.00	45.64 ± 00.20	45.64 ± 00.20	45.64 ± 00.20	06.97 ± 04.53	45.64 ± 00.20
	O%	100.00 ± 00.00	$\mathbf{00.00 \pm 00.00}$	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00
	I%	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00
CARCINOGENESIS	F_1	00.00 ± 00.00	$\mathbf{70.51 \pm 03.10}$	70.46 ± 03.09	$\mathbf{70.51 \pm 03.10}$	71.48 ± 08.34	66.26 ± 13.26
	M%	00.00 ± 00.00	$\mathbf{54.36 \pm 00.20}$	54.47 ± 03.70	54.36 ± 00.20	63.42 ± 10.34	49.23 ± 14.82
	C%	00.00 ± 00.00	45.64 ± 00.20	45.53 ± 03.70	45.64 ± 00.20	36.58 ± 10.34	40.58 ± 14.87
	O%	100.00 ± 00.00	$\mathbf{00.00 \pm 00.00}$	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	09.09 ± 28.75
	I%	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00	00.00 ± 00.00

4.2 Outcomes

Tab. 2 reports the results of the experiments (the improvements due to the new refinement operators are reported by using bold font style). In general, when the refinement operator proposed in [3, 4] and TDTs are considered in the experiments, we observed a large omission rate for each ontology. This results can be explained by the difficulty of TDTs to recognize negative instances, likely due to the lack of useful disjointness axioms and the Open World Assumption.

Concerning the experiments with LYMPH ontology, we noticed an improvement w.r.t. the original version of the learning algorithms when we resort to the RHO operator and the regularizer term. The improvement of the match rate and the F-measure in the case of TDTs were really prominent (these improvements were around 28% and 82%, respectively). In this case the models were competitive w.r.t. the concepts induced through CELOE and ELTL DISJUNCTIVE. Thanks to the new refinement operator, each tree contained concept descriptions that allowed to discern positive instances and to recognize the negative examples. In addition, we noticed that for this learning problem, a larger number of positive instances were available and this could affect the quality of the trees.

As regards the NTN ontology and the employment of the two refinement operators, the TDTs and ETDTs improved the performance w.r.t. the original versions of the mode only thanks to the RHO operator and the regularizer term. Also, in this case various missing values were found, like the experiments with LYMPH, and the uncertain membership was assigned to test individuals. Consequently, the F-measure was very low: it was only 30%. On the other hand, resorting to the ETDTs with the original refinement operator improved the F-measure, which was 40% , and partially mitigated the number of omission cases thanks to the strategy employed for dealing with the missing values. In this case a large number of negative instances have been predicted. With the RHO operator and the regularizer, we observed a significant improvement of the match rate for TDTs, around 70%. For ETDTs the increase was more limited, but it was still good enough:

it was about of 14%. The improvement in terms of F-measure was very large: it was around 70% for TDTs and 60% for ETDTs. This result can be explained by the possibility to set various parameters for RHO operator in order to be fitted w.r.t. the specific learning problem, for instance by setting opportunely the use of data properties. Thanks to the integration of the refinement operator the results are better than the ones obtained by exploiting ELTL DISJUNCTIVE, which induced very poor concept descriptions, and similar to the ones obtained by resorting to CELOE. Finally, in the case of MUTAGENESIS and CARCINOGENESIS ontology, we observed a bad result for the experiments with the original version of TDTs: all test individuals were classified as having an uncertain membership. This was likely due to the expressiveness of the ontology, which is really limited for the refinement operator employed in this experiment. As explained in Sect. 2, the latter considers only concept names and existential or universal restriction obtained from roles that can be found in a knowledge base. But the expressiveness of MUTAGENESIS did not allowed to find this kind of candidate concepts. Besides, no disjointness axiom was found in this ontology. This limit of TDTs is broader when we compared their predictiveness with the one of the original ETDTs: these models were able to reduce the number of omission cases and improve the F-measure. With the RHO operator, the performance of TDTs improved significantly. In fact, the match rate and the F-measure are comparable to the ones obtained via ETDTs, by applying both the original refinement operator and RHO. Similarly to the case of NTN, the performance is better than or as well as the ones obtained through ELTL DISJUNCTIVE although it was worse than the performance of CELOE. This may be due to the fact that CELOE is an accuracy-driven method for inducing concepts which exploits the most promising description for classifying individuals. Conversely, the greedy algorithm employed for growing trees could yield sub-optimal solutions.

5 Conclusions and Extensions

In this work, we integrated the refinements operators available in DL-Learner into the learning algorithms for inducing Terminological Decision Trees and Evidential Terminological Decision Trees. We also proposed to modify the heuristic for selecting the best concept in order to take into account the similarity between a specialization and the concept installed into the father node. An empirical evaluation showed that by modifying the learning algorithms, the resulting models have better performance w.r.t. the original version. Besides the new models can fit with a lower expressivity than the one considered by the original refinement operator. Unfortunately, for some learning problems, the tree models did not outperform other methods proposed in the literature. This work is still preliminary and it can be extended along various directions. Firstly, we can extend the comparison by exploiting further refinement operators and further regularizer terms. In addition, we plan to extend the empirical evaluation by considering also further ontologies and further learning problems in order to investigate the correlation existing between the learning algorithms, the refinement operators and the expressiveness of the ontologies considered in the experiments.

Acknowledgements

This work fulfills the objectives of the PON 02005633489339 project "Puglia@Service - Internet-based Service Engineering enabling Smart Territory structural development" funded by the Italian Ministry of University and Research (MIUR).

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook. 2nd edn. Cambridge University Press (2007)
2. Rettinger, A., Lösch, U., Tresp, V., d'Amato, C., Fanizzi, N.: Mining the semantic web - statistical learning for next generation knowledge bases. Data Min. Knowl. Discov. **24** (2012) 613–662
3. Fanizzi, N., d'Amato, C., Esposito, F.: Induction of concepts in web ontologies through terminological decision trees. In Balcázar, J., et al., eds.: Proceedings of ECML/PKDD2010. Volume 6321 of LNAI., Springer (2010) 442–457
4. Rizzo, G., d'Amato, C., Fanizzi, N., Esposito, F.: Towards evidence-based terminological decision trees. In Laurent, A., et al., eds.: Information Processing and Management of Uncertainty in Knowledge-Based Systems - 15th International Conference, IPMU 2014 Proceedings, Part I. Volume 442 of Communications in Computer and Information Science., Springer (2014) 36–45
5. Rizzo, G., d'Amato, C., Fanizzi, N.: On the effectiveness of evidence-based terminological decision trees. In Esposito, F., Pivert, O., Hacid, M., Ras, Z.W., Ferilli, S., eds.: Foundations of Intelligent Systems - 22nd International Symposium, ISMIS 2015, Lyon, France, October 21-23, 2015, Proceedings. Volume 9384 of Lecture Notes in Computer Science., Springer (2015) 139–149
6. Fanizzi, N., d'Amato, C., Esposito, F.: DL-FOIL concept learning in description logics. In Zelezný, F., Lavrac, N., eds.: Inductive Logic Programming, 18th International Conference, ILP 2008, Prague, Czech Republic, September 10-12, 2008, Proceedings. Volume 5194 of Lecture Notes in Computer Science., Springer (2008) 107–121
7. Lehmann, J.: Dl-learner: Learning concepts in description logics. Journal of Machine Learning Research (JMLR) **10** (2009) 2639–2642
8. Lehmann, J., Auer, S., Bühmann, L., Tramp, S.: Class expression learning for ontology engineering. J. Web Sem. (2011) 71–81
9. Lehmann, J., Haase, C.: Ideal downward refinement in the $\mathcal{EL}$ description logic. In Raedt, L.D., ed.: Inductive Logic Programming, 19th International Conference, ILP 2009. Revised Papers. Volume 5989 of Lecture Notes in Computer Science., Springer (2009) 73–87
10. Klir, J.: Uncertainty and Information. Wiley (2006)
11. Lehmann, J., Hitzler, P.: Concept learning in description logics using refinement operators. Machine Learning **78** (2010) 203–250
12. Smarandache, F., Han, D., Martin, A.: Comparative study of contradiction measures in the theory of belief functions. In: 15th International Conference on Information Fusion, FUSION 2012, Singapore, July 9-12, 2012. (2012) 271–277
13. Sentz, K., Ferson, S.: Combination of evidence in Dempster-Shafer theory. Volume 4015. Citeseer (2002)
14. Dubois, D., Prade, H.: On the combination of evidence in various mathematical frameworks. In Flamm, J., Luisi, T., eds.: Reliability Data Collection and Analysis. Volume 3 of Eurocourses. Springer Netherlands (1992) 213–241
15. Deng, H., Runger, G.C.: Feature selection via regularized trees. In: The 2012 International Joint Conference on Neural Networks (IJCNN), 2012, IEEE (2012)