# Mining the Semantic Web with Machine Learning: main issues that need to be taken into account

Claudia d'Amato

*Department of Computer Science*
*University of Bari*

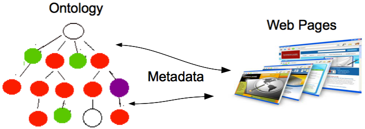
Reasoning Web Summer School 2021
Virtual, September 9, 2021

# Contents

# Semantic Web and Ontologies

**Semantic Web** (SW) **goal:** making data on the Web machine understandable [Berners-Lee *et al.*, 2001]

- ontologies play a key role acting as a *shared vocabulary for assigning data semantics*

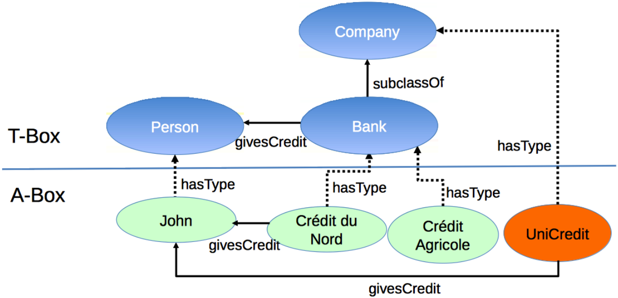



Examples of existing real ontologies

- Schema.org
- Gene Ontology
- Foundational Model of Anatomy ontology
- Financial Industry Business Ontology (by OMG Finance Domain Task Force)
- GoodRelations
- ...

OWL standard language ⇒ **Description Logics** (DLs) theoretical foundation

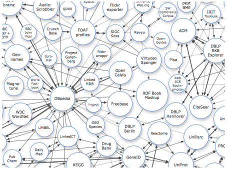
*Ontologies equipped with* deductive reasoning capabilities ⇒ allowing to make explicit, knowledge that is implicit within them




**Deduction:**
"Crédit du Nord",
"Crédit Agricole"
are also `Company`

# The Web of Data

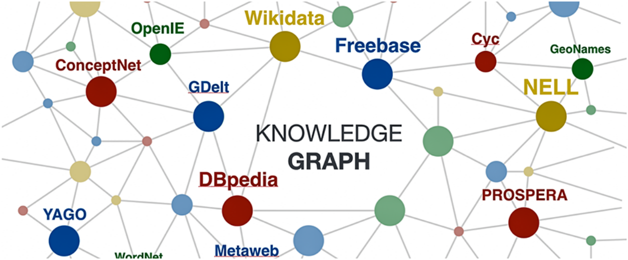
- Progressive increasing amount of annotated and interlinked data on the Web
- **Web of Data** global scale interlinking ontologies and data [Shadbolt *et al.*, 2006]




- *Linked Data*: rules for making easier and easier publishing, linking and sharing data on the Web [Berners-Lee, 2006]
- *Linked Open Data*[1] public openness and availability of larger and larger datasets ⇒ relevance and centrality of **DBpedia**[2] as a driving force

[1] https://lod-cloud.net/versions/latest/lod-cloud.svg

[2] http://dbpedia.org



[3]

Open KG
online with content freely accessible

- BabelNet
- DBpedia
- Freebase
- Wikidata
- YAGO
- ....

Enterprise KG
for commercial usage

- Google
- Amazon
- Facebook
- LinkedIn
- Microsoft
- ....

[3] picture from https://www.csee.umbc.edu/courses/graduate/691/fall19/07/

## Applications

- e-Commerce
- Semantic Search
- Fact Checking
- Personalization
- Recommendation
- Medical decision support system
- Question Answering
- Machine Translation
- ...

## Research Areas

- Information Extraction
- Natural Language Processing
- Machine Learnig (ML)
- Knowledge Representation
- Web
- Robotics
- ...

## Knowledge Graph: Definition [Hogan *et al.*, 2021]

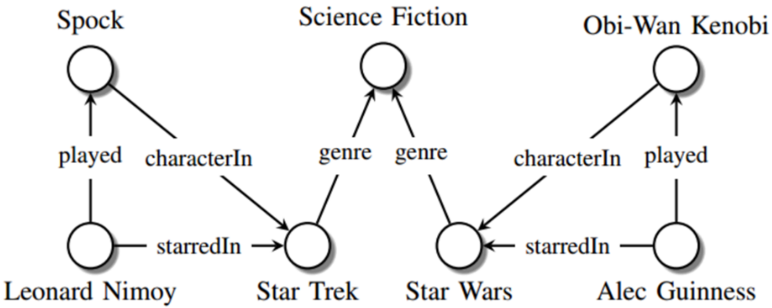
A graph of data intended to convey knowledge of the real world

- conforming to a graph-based data model
- nodes represent entities of interest
- edges represent potentially different relations between these entities
- data graph potentially enhanced with schema

## KGs: Main Features

- *ontologies* employed to define and reason about the semantics of nodes and edges
- RDF, RDFS, OWL representation languages will be assumed
- grounded on the Open World Assumption (OWA)
- very large data collections

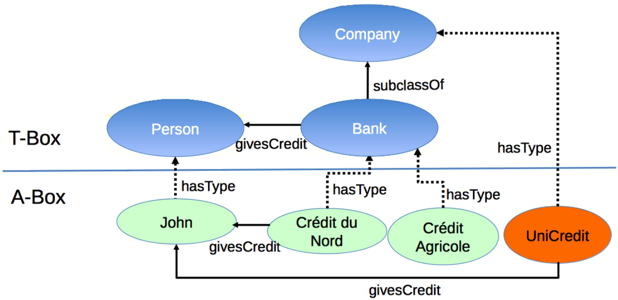
# Knowledge Graph: Example




Source: Maximilian Nickel et al. A Review of Relational Machine Learning for Knowledge Graphs: From Multi-Relational Link Prediction to Automated Knowledge Graph Construction

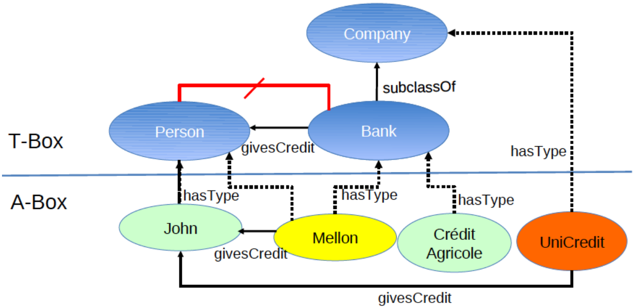
# Issues

- KG suffer of *incompleteness* and *noise*
  - e.g. missing links, wrong links
  - since often result from a complex building process
- Ontologies and assertions can be out-of-sync
  - resulting incomplete, noisy and sometimes inconsistent wrt the actual usage of the conceptual vocabulary in the assertions
- Reasoning cannot be performed or may return counterintuitive results



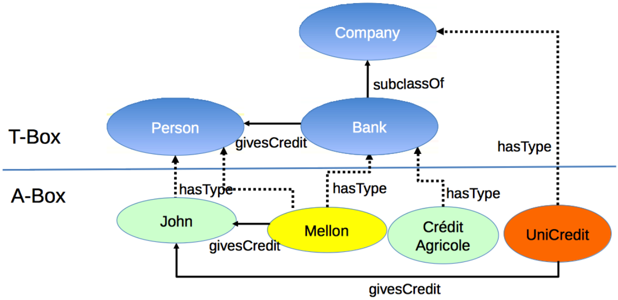

**Incompleteness**

UniCredit is a Bank




**Inconsistency**

Mellon cannot be a `Person` **and** a `Bank`




**Noise**

Person $\equiv \neg$Bank missing

Machine Learning methods adopted to discover new/additional knowledge by exploiting *the evidence coming from the data* [d'Amato *et al.*, 2010; d'Amato, 2020]

**Machine Learning:** the study of systems that improve their behavior over time with experience [Mitchell, 1997; MacKay, 2002; Flach, 2012; Murphy, 2012]

experience:

- interactions with the world
- set of *observations* or *examples*
- internal states and processes

**ML Approaches:** [Luger, 2005]

- symbol-based
- numeric / connectionist / neurally inspired

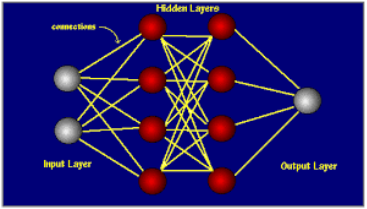
# Symbol-Based Learning

- uses symbols for representing entities and relationships of a domain (observations/examples)
- infer novel, valid and useful *generalizations* of examples
  - that provide new *insights* into the data/examples
  - are ideally readily *interpretable* by the user
- by *searching* thought possible generalizations expressed with symbols

**Induction** typically adopted

# Neurally Inspired Learning

- represents knowledge as patterns of activity in networks of small, individual processing units
  - needs to **encode knowledge into numerical quantities** in the network
- learns by *modifying* / adapting the network structure and weights in response to incoming (training) data
  - *does not learn by adding representation to the KB*



# Induction vs. Deduction

**Deduction (Truth preserving)**

Given:

- a set of general axioms
- a proof procedure

Draw:

- *correct and certain* conclusions

**Induction (Falsity preserving)**

Given:

- a set of examples

Determine:

- a *possible/plausible* generalization covering
  - the given examples/observations
  - new and not previously observed examples

## Supervised Learning (Learning from examples)

- Given a training set $\{(\mathbf{x_1}, y_1), \ldots (\mathbf{x_n}, y_n)\}$ where $\mathbf{x_i}$ are input examples and $y_i$ the desired output, learn an unknown function $f$ such that $f(x) = y$ for new examples
  - $y$ having discrete values $\Rightarrow$ *Classification Problem*
  - $y$ having continuos values $\Rightarrow$ *Regression Problem*
  - $y$ having a probability value $\Rightarrow$ *Probability Estimation Problem*
- Supervised Concept Learning:
  - Given a training set of positive and negative examples for a concept,
  - *construct* a *description* that will accurately classify whether future examples are positive or negative.

## Unsupervised Learning (Learning from Observations)

- Given a set of observations $\{\mathbf{x_1}, \ldots \mathbf{x_n}\}$
  - discover hidden patterns in the data $\Rightarrow$ *Discovery*
  - for a concept/class/category, construct a description that is able to determine if a (new) example is an instance of the concept (positive example) or not (called negative example). $\Rightarrow$ *Concept Learning*
  - assess groups of similar data items $\Rightarrow$ *Clustering*

## Semi-supervised learning

- is halfway between supervised and unsupervised learning
- training data is built up by both few labeled (i.e. with the desired output) and unlabeled data
- both kinds of data are used for solving the learning tasks (almost the same tasks as for the case of supervised learning)

# Machine Learning & Semantic Web

**Symbol-based methods**

- able to exploit background knowledge and (deductive) reasoning capabilities
- limited in scalability

$\Downarrow$

**Ontology Mining**

- *All activities that allow for discovering hidden knowledge from ontological KBs*

**Numeric-based methods**

- highly scalable
- schema level information and reasoning capabilities almost disregarded

$\Downarrow$

**Knowledge Graph Refinement**

- *Link Prediction*: predicts missing links between entities
- *Triple Classification*: assesses statement correctness in a KG

[d'Amato, 2020]

# Symbol-based Methods for Ontology Mining

## Ontology Mining Tasks

- Instance Retrieval (Instance Level)
- Ontology Enrichment (Schema Level)

**from an inductive perspective**

## Ontology Mining Tasks

- **Instance Retrieval (Instance Level)**
- Ontology Enrichment (Schema Level)

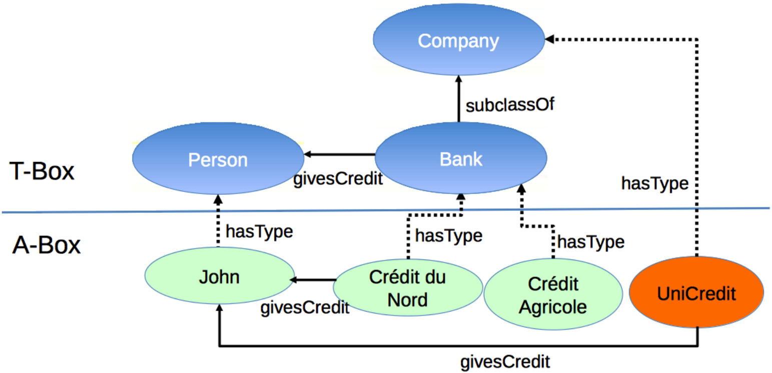
**from an inductive perspective**

# Instance Retrieval as a Classification Problem

# Introducing Instance Retrieval I

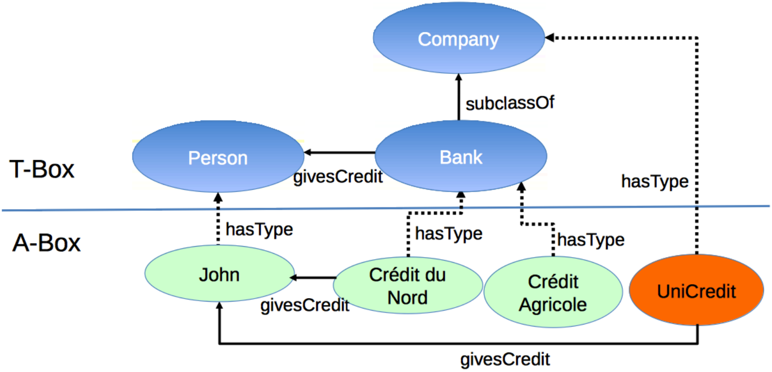
*Instance Retrieval* → Finding the extension of a query concept

- Instance Retrieval (Bank) = {"Crédit du Nord","Crédit Agricole"}



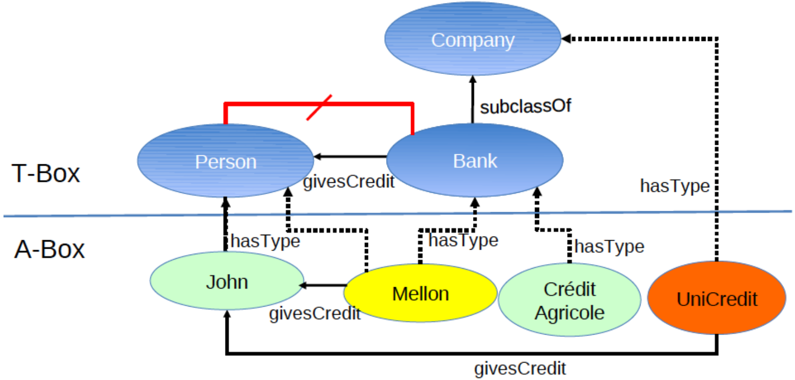
# Introducing Instance Retrieval I

**Problem:** Instance Retrieval in incomplete/inconsistent/noisy ontologies



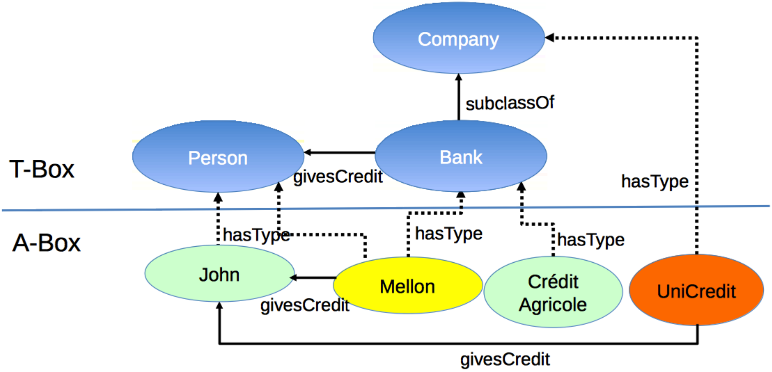
# Introducing Instance Retrieval II

**Problem:** Instance Retrieval in incomplete/inconsistent/noisy ontologies



# Introducing Instance Retrieval III

**Problem:** Instance Retrieval in incomplete/inconsistent/noisy ontologies



### Idea

**Casting** the problem as a **classification problem**

*assess the class membership of individuals in a DL KB w.r.t. the query concept*

Similarity-based methods mostly adopted $\Rightarrow$ **efficient and noise tolerant**

**Issues: State of art classification methods cannot be** straightforwardly applied

- generally applied to *feature vector* representation
  $\rightarrow$ *upgrade DL expressive representations*
- implicit *Closed World Assumption* made in ML
  $\rightarrow$ *cope with the Open World Assumption made in DLs*
- classes considered as *disjoint*
  $\rightarrow$ *cannot assume disjointness of all concepts*

## Adopted Solutions:

- Defined new semantic similarity measures for DL representations [d'Amato, 2007]
  - to cope with the high expressive power of DLs
  - to deal with the semantics of the compared objects (concepts, individuals, ontologies)
  - to convey the underlying semantics of KB
- Formalized a set of criteria that a similarity function has to satisfy for being defined *semantic* [d'Amato *et al.*, 2008a]
- Definition of the classification problem taking into account OWA
- Multi-class classification problem decomposed into a set a smaller classification problems

### Definition (Problem Definition)

**Given:**

- a populated ontological knowledge base $KB = (\mathcal{T}, \mathcal{A})$
- a query concept $Q$
- a training set with $\{+1, -1, 0\}$ as target values

**Learn a classification function $f$ such that:** $\forall a \in Ind(\mathcal{A})$ :

- $f(a) = +1$ if $a$ is instance of $Q$
- $f(a) = -1$ if $a$ is instance of $\neg Q$
- $f(a) = 0$ otherwise (unknown classification because of OWA)

## Dual Problem

- given an individual $a \in Ind(\mathcal{A})$, tell concepts $C_1, \ldots, C_k$ in $KB$ it belongs to
- the multi-class classification problem is *decomposed* into a set of *ternary classification problems* (one per target concept)

# Developed methods

**Pioneering the Problem**

- relational K-NN for DL KBs [d'Amato *et al.*, 2008b]

**Improving the efficiency**

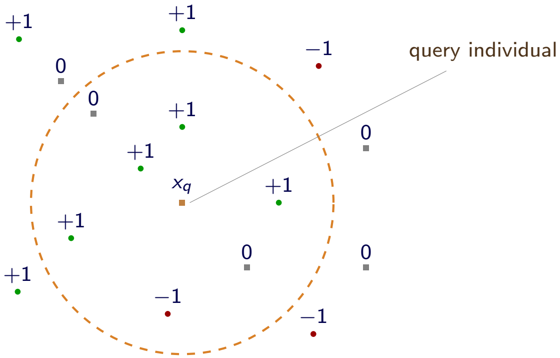
- kernel functions for kernel methods to be applied to DLs KBs [Fanizzi and d'Amato, 2006; Fanizzi *et al.*, 2012a; Bloehdorn and Sure, 2007]

**Scaling on large datasets**

- Statistical Relational Learning methods for large scale and data sparseness [Huang *et al.*, 2010; Minervini *et al.*, 2015]

# Example: Nearest Neighbor Classification

query concept: `Bank` $k = 7$
target values standing for the class values: $\{+1, 0, -1\}$

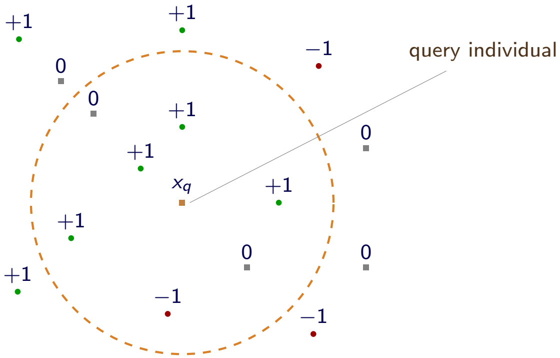



$class(x_q) \leftarrow ?$

# Example: Nearest Neighbor Classification

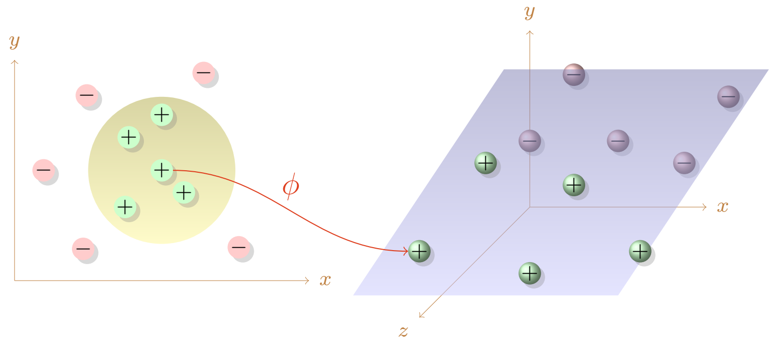
query concept: `Bank` $k = 7$

target values standing for the class values: $\{+1, 0, -1\}$




$class(x_q) \leftarrow +1$

# Example: Kernel Method Classification



# On evaluating the Classifier

**Problem:** How evaluating classification results?

- **Inductive Classification compared with a standard reasoner** (PELLET)
- Query concepts from ontologies publicly available considered
- Registered *mismatches*: Induction: $\{+1, -1\}$ - Deduction: no results
- **Evaluated as mistake if precision and recall were used** while it could turn out to be a correct inference when judged by a human

**Defined new metrics** *to distinguish induced assertions from mistakes*

| | | REASONER | | |
|---|---|---|---|---|
| | | +1 | 0 | -1 |
| INDUCTIVE CLASSIFIER | +1 | *M* | *I* | *C* |
| | 0 | *O* | *M* | *O* |
| | -1 | *C* | *I* | *M* |

*M* **Match Rate**
*C* **Commission Error Rate**
*O* **Ommission Error Rate**
*I* **Induction Rate**

# Lesson Learnt from experiments

- *Commission error* almost zero on average
- *Omission error rate* very low and only in some cases
  - Not null for ontologies in which disjoint axioms are missing
- *Induction Rate* not zero
  - **new knowledge (not logically derivable) induced** $\Rightarrow$ can be used for *semi-automatizing the ontology population task*
  - induced knowledge $\Rightarrow$ *individuals are instances of many concepts* and *homogeneously spread* w.r.t. the several concepts.

| | match | commission | omission | induction |
|---|---|---|---|---|
| SWM | 97.5 ± 3.2 | 0.0 ± 0.0 | 2.2 ± 3.1 | 0.3 ± 1.2 |
| LUBM | 99.5 ± 0.7 | 0.0 ± 0.0 | 0.5 ± 0.7 | 0.0 ± 0.0 |
| NTN | 97.5 ± 1.9 | 0.6 ± 0.7 | 1.3 ± 1.4 | 0.6 ± 1.7 |
| FINANCIAL | 99.7 ± 0.2 | 0.0 ± 0.0 | 0.0 ± 0.0 | 0.2 ± 0.2 |

# Research Directions to Investigate Further

- Multi-Label Classification
  - individuals can be instance of more than one concept at the same time [Melo and Paulheim, 2019; Peixoto *et al.*, 2016]
- Hierarchical Classification
  - Particularly appropriate for type prediction [Melo *et al.*, 2016, 2017]
- Ensemble methods
  - only boosting has been preliminarily applied [Rizzo *et al.*, 2015a; Fanizzi *et al.*, 2019]
- Regression
  - to be exploited for predicting missing values of datatypes properties [Fanizzi *et al.*, 2012b; Rizzo *et al.*, 2016]

## Ontology Mining Tasks

- Instance Retrieval (Instance Level)
- **Ontology Enrichment (Schema Level)**

**from an inductive perspective**

# Ontology enrichment as a Concept Learning Problem

# On Learning Concept Descriptions I

**Goal:** Learning descriptions for a given concept name / expression

$$\textit{Example}: \quad \mathsf{Man} \equiv \mathsf{Human} \sqcap \mathsf{Male}$$

**Question:** How to learn concept descriptions automatically, given a set of individuals?

**Idea**

Regarding the problem as a *supervised concept learning* task

Supervised Concept Learning:

- Given a training set of positive and negative examples for a concept,
- *construct* a *description* that will accurately classify whether future examples are positive or negative.

# On Learning Concept Descriptions II

**Definition (Problem Definition)**

- *Given*
  - the KB $\mathcal{K}$ as a background knowledge
  - a subset *pos* of individuals as positive examples of $C$
  - a subset *neg* of individuals as negative examples of $C$
- *Learn*
  - a DL concept description $D$ so that
  - the individuals in *pos* are instances of $D$ while those in *neg* are not

# The Learning Process:
# Learning as Search

# How Does Relational Learning Work?

Symbolic ML techniques essentially search a space of possible hypothesis $\mathcal{L}_h$ (e.g. patterns, models, regularities) [De Raedt, 2008]

- Depending on the task, different search algorithms and principles apply
  - *complete search* strategy applicable
  - *heuristic search* method (e.g. *hill climbing*)
- easy way: *generate-and-test algorithm*
  - naïve and inefficient

# A Generate-and-Test Algorithm

A (trivial) algorithm based on a *generate-and-test* technique is the **enumeration algorithm**

- for each possible hypothesis $h$ checks if $h$ satisfies a given quality criterion $Q$ wrt the data $D$

```
for each h ∈ 𝓛_h do
    if Q(h, D) = true then
        output h
    end if
end for
```

*Properties*

- whenever a solution exists, the enumeration algorithm will find it
- it can only be applied if the hypotheses language $\mathcal{L}_h$ is *enumerable*
- the algorithm searches the *whole* space $\rightarrow$ inefficient
  - it is advantageous to *structure* the search space, according to *generality* allowing for its *pruning*

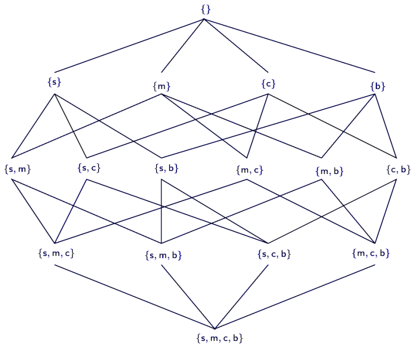
Usually *logical entailment* used as for generality relation

- a more general hypothesis logically *entails* the more specific one
- a more specific hypothesis is a *logical consequence* of the more general one

**Definition (generality)**

Let $h_1, h_2 \in \mathcal{L}_h$. Hypothesis $h_1$ is *more general than* (or equivalent) hypothesis $h_2$ , $h_1 \preceq h_2$, iff all examples covered by $h_2$ are also covered by $h_1$, i.e., $c(h_2) \subseteq c(h_1)$

- We also say that
  - $h_2$ is a *specialization* of $h_1$
  - $h_1$ is a *generalization* of $h_2$
- $h_1$ is a **proper generalization** of $h_2$, $h_1 \prec h_2$
  when $h_1 \preceq h_2$
  and $h_1$ covers examples not covered by $h_2$




**Space traversed in:**

- a *general-to-specific* strategy:
  - the algorithm starts from the *most general hypothesis*
  - then repeatedly specializes mapping hypothesis /patterns onto a set of specializations
- a *specific-to-general* strategy

Notice that the $\preceq$ is transitive and reflexive; $\rightarrow$ it is a *quasi-order*

- not anti-symmetric since *there may exist several hypotheses that cover* exactly the same set of examples: *syntactic variants*
  - undesirable: they introduce redundancies in the search space

# Monotonicity I

The generality relation imposes a useful structure on the search space provided that the quality criterion involves some properties:

**Definition (monotonicity of the criteria)**

A quality criterion $Q$ is **monotonic** iff

$$\forall s, g \in \mathcal{L}_h, \forall D \subseteq \mathcal{L}_e : (g \preceq s) \wedge \; Q(g, D) \rightarrow Q(s, D)$$

It is **anti-monotonic** iff

$$\forall s, g \in \mathcal{L}_h, \forall D \subseteq \mathcal{L}_e : (g \preceq s) \wedge \; Q(s, D) \rightarrow Q(g, D)$$

# Monotonicity II

Properties that directly follow from the definitions of monotonicity and anti-monotonicity:

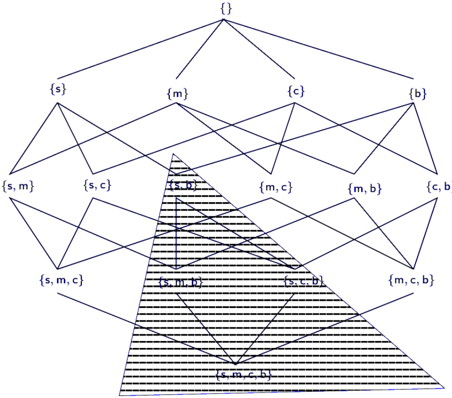
**Property (prune generalizations)**

*If a hypothesis $h$ does not satisfy a monotonic quality criterion then none of its generalizations will*

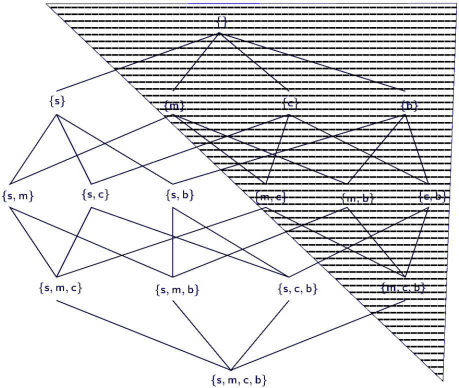
**Property (prune specializations)**

*If a hypothesis $h$ does not satisfy an anti-monotonic quality criterion then none of its specializations will*

# Monotonicity III




prune specializations




prune generalizations

# Refinement Operators I

**How** can be the search space $\mathcal{L}_h$ traversed?

Many ML algorithms are based on **refinement operators**

- generating sets of specializations (or generalizations) of given hypotheses

## Definition

A **generalization operator** $\rho_g \colon \mathcal{L}_h \to 2^{\mathcal{L}_h}$ is a function such that

$$\forall h \in \mathcal{L}_h \colon \rho_g(h) \subseteq \{h' \in \mathcal{L}_h \mid h' \preceq h\}$$

Dually, a **specialization operator** $\rho_s \colon \mathcal{L}_h \to 2^{\mathcal{L}_h}$ is a function such that

$$\forall h \in \mathcal{L}_h \colon \rho_s(h) \subseteq \{h' \in \mathcal{L}_h \mid h \preceq h'\}$$

# Refinement Operators II

## Properties

defined for specialization op's (corresponding definitions for generalization op's easily obtained)

- $\rho$ is an **ideal operator** for $\mathcal{L}_h$ iff
  $\forall h \in \mathcal{L}_h \colon \rho(h) = \min(\{h' \in \mathcal{L}_h \mid h \prec h'\})$
  - it returns all children for a node in the Hasse diagram
    - proper refinements, not a syntactic variant of the original hypothesis
  - often are used in *heuristic search* algorithms
- $\rho$ is an **optimal operator** for $L_h$ iff for all $h \in \mathcal{L}_h$ there exists exactly *one* sequence of hypotheses $\top = h_0, h_1, \ldots, h_n = h \in \mathcal{L}_h$ such that $h_i \in \rho(h_{i-1})$ for all $i$
  - used in *complete search* algorithms
- An operator for which there exists *at least* one sequence from $\top$ to any $h \in \mathcal{L}_h$ is called **complete**
- An operator for which there exists *at most* one such sequence is **non-redundant**

# A Generic Learning Algorithm I

Adapting the enumeration algorithm to employ the refinement operators:

```
Queue ← Init
Th ← ∅
while not Stop do
    Delete h from Queue
    if Q(h, D) then
        Th ← Th ∪ {h}
        Queue ← Queue ∪ ρ(h)
    end if
    Queue ← Prune(Queue)
    end while
return Th
```

# A Generic Learning Algorithm II

Observations. many parameters determining the behavior

- *Init* determines the *starting point* of the search algorithm
  - The initialization may yield one or more initial hypotheses
  - Most algorithms start either at $\top$ and only specialize (the so-called general-to-specific systems), or at $\bot$ and only generalize (the specific-to-general systems)
- *Delete* determines the *search strategy*
  - *first-in-first-out*: breadth-first search
  - *last-in-first-out*: depth-first search
  - *best hypothesis* (according to some criterion or heuristic): best-first algorithm
- $\rho$ determines the size and nature of the *refinement steps* through the search space
- *Stop* determines when the algorithm *halts*

# A Generic Learning Algorithm III

- Some algorithms compute all elements, $k$ elements or an approximation of an element satisfying $Q$
  - if all elements are desired, $Stop$ equals $Queue = \emptyset$
  - when $k$ elements are sought, it is $|Th| = k$
- Some algorithms $Prune$ candidate hypotheses from $Queue$
  - *heuristic pruning* prunes away parts of the search space that appear to be uninteresting
  - *sound pruning* prunes away parts of the search space that cannot contain solutions
- As with other search algorithms in AI:
  - *complete* algorithms compute all elements of $Th(Q, D, \mathcal{L}_h)$
  - *heuristic* algorithms aim at computing one or a few hypotheses that score best w.r.t. a given heuristic
    - not guaranteeing that the best hypotheses are found

# Concept Learning in Description Logics

# DL Concept Learning – Problem Definition I

**given**
- a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$
- a target concept $C$
- a set of training instances partitioned as examples and counterexamples $\mathbf{E} = \mathbf{E}_{+} \cup \mathbf{E}_{-}$ for $C$

**find** a description $D$ for $C$ generalizing $\mathbf{E}$, $C \equiv D$, that maximizes the *accuracy* w.r.t. the positive and negative examples

**Possible Issues:**

- *Negative examples*: ML grounded on CWA, DLs based on OWA
  - Learning from positive examples only if negative examples missing
- Suitable *refinement operators* needed
- *Evaluating results*: metrics, unbalanced setting

# DL Concept Learning – Problem Definition II

**Accuracy**

$D$ correctly *entails* at least $(1-\epsilon)|\mathbf{E}|$ of the assertions on examples regarding their membership to $C$:
$\forall e \in \mathbf{E}_+ : \ \mathcal{K} \sqcup \{D\} \models C(e)$ and
$\forall e \in \mathbf{E}_- : \ \mathcal{K} \sqcup \{D\} \not\models C(e)$

stronger alternative:
$\forall e \in \mathbf{E}_- : \ \mathcal{K} \sqcup \{D\} \models \neg C(e)$

Variant: separate $\epsilon_+$ and $\epsilon_-$

# Refinement Operators

Randomized recursive **refinement operator** $\rho$

$$C' \in \rho(C)$$

1. $C' = C \sqcap A$
2. $C' = C \sqcap \neg A$
3. $C' = C \sqcap \forall R.\top$
4. $C' = C \sqcap \exists R.\top$
5. $C' = C_1 \sqcap \cdots \sqcap B \sqcap \cdots \sqcap C_n$
   if $C = C_1 \sqcap \cdots \sqcap A \sqcap \cdots \sqcap C_n$ and $B \sqsubseteq A$
6. $C' = C_1 \sqcap \cdots \sqcap \neg B \sqcap \cdots \sqcap C_n$
   if $C = C_1 \sqcap \cdots \sqcap \neg A \sqcap \cdots \sqcap C_n$ and $A \sqsupseteq B$
7. $C' = C_1 \sqcap \cdots \sqcap \exists R.D \sqcap \cdots \sqcap C_n$
   if $C = C_1 \sqcap \cdots \sqcap \exists R.E \sqcap \cdots \sqcap C_n$ and $D \in \rho(E)$
8. $C' = C_1 \sqcap \cdots \sqcap \forall R.D \sqcap \cdots \sqcap C_n$
   if $C = C_1 \sqcap \cdots \sqcap \forall R.E \sqcap \cdots \sqcap C_n$ and $D \in \rho(E)$

# Developed Methods for Supervised Concept Learning

- **Separate-and-conquer approach**
  - YinYang [Iannone *et al.*, 2007]
  - DL-FOIL [Fanizzi *et al.*, 2008, 2018]
  - DL-Learner [Lehmann and Hitzler, 2010]
  - CELOE [Lehmann *et al.*, 2011]
  - DL-FOCL [Rizzo *et al.*, 2020]
- **Divide-and-conquer approach**
  - TermiTIS [Fanizzi *et al.*, 2010]
  - PARCEL [Tran *et al.*, 2012]
  - SPACEL [Tran *et al.*, 2017]
  - TERMITIS – EXTENSIONS
    - Pruning Methods [Rizzo *et al.*, 2017b,a] - simplify complexity & avoid overfitting
    - *Terminological Random Forests* TRFs [Rizzo *et al.*, 2015a] - tackling also the *class-imbalance* problem
    - Evidential TDTs and TRFs [Rizzo *et al.*, 2018, 2015b] - based on the *Dempster-Shafer Theory*(DST): a general framework for reasoning with uncertainty

# DL-FOIL I

*Problem*: simple *generate-and-test* algorithms may be inefficient

**DL-FOIL** adopt a heuristic *sequential covering* algorithm [Fanizzi *et al.*, 2008; Fanizzi, 2011]

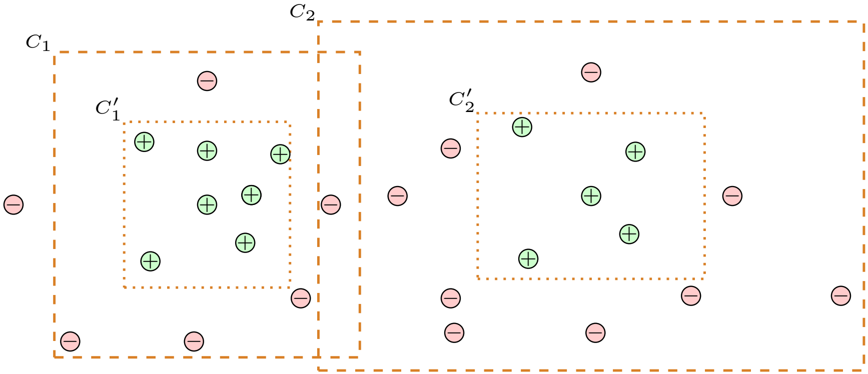
*general-to-specific search*

- starting from $\top$
- **repeat** (cover as many positives as possible)
  - **if** non positives are covered
  - **repeat**
    - find heuristically the best refinement
      (not to cover them yet still covering as many positives as possible)
    - add refinement as a disjunct partial def.

  **until** only positives covered

  **until** all positives covered

# DL-FOIL II




$C_1$ = MasterStudent    $C'_1$ = MasterStudent $\sqcap$ $\exists$worskIn.$\top$

$C_2$ = BachelorStudent    $C'_2$ = BachelorStudent $\sqcap$ $\exists$worskIn.$\top$

# DL-FOIL III

Heuristic function: **Gain**

$$g(D_0, D_1) = p_1 \cdot \left[\log \frac{p_1}{p_1 + n_1 + u_1} - \log \frac{p_0}{p_0 + n_0 + u_0}\right]$$

where

- $p_1|n_1|u_1$ number of exs covered by the specialized def. $D_1$
- $p_0|n_0|u_0$ number of exs covered by the former (partial) def. $D_0$

$+$ correction via *Laplace smoothing*

# On Evaluating the Learnt Concept Descriptions

- Publicly available ontologies considered
- A number (30) of satisfiable randomly generated concepts considered
- Positive and negative examples collected for each concept by using a deductive reasoner
- Running concept learning on the collected positive and negative examples
- Inductive classification performed on the learnt concept descriptions

| ontology | match rate | commission error rate | omission error rate | induction rate |
|---|---|---|---|---|
| BioPax | **76.9** ± 15.7 | **19.7** ± 15.9 | **7.0** ± 20.0 | **7.5** ± 23.7 |
| NTN | **78.0** ± 19.2 | **16.1** ± 4.0 | **6.4** ± 8.1 | **14.0** ± 10.1 |
| Financial | **75.5** ± 20.8 | **16.1** ± 12.8 | **4.5** ± 5.1 | **3.7** ± 7.9 |

# Examples of Learned Descriptions with DL-FOIL

BioPax

*induced:*

`Or( And( physicalEntity protein) dataSource)`

*original:*

```
Or( And( And( dataSource externalReferenceUtilityClass)
ForAll(ORGANISM ForAll(CONTROLLED phys icalInteraction)))
protein)
```

NTN

*induced:*

`Or( EvilSupernaturalBeing Not(God))`

*original:*

`Not(God)`

Financial

*induced:*

`Or( Not(Finished) NotPaidFinishedLoan Weekly)`

*original:*

`Or( LoanPayment Not(NoProblemsFinishedLoan))`

# Lesson Learnt from Experiments

- Relatively small ontological KBs adopted $\Rightarrow$ *scalability needs to be improved*
- Suitable concept descriptions learned $\Rightarrow$ *validation by expert recommended for adding axioms to the KB*
  - approximated descriptions may be learned depending of the threshold

# Ontology enrichment as a Disjointness Axioms Learning Problem

## A fine grained schema level information can bring better insight of the data

Disjointness axioms often missing

Problems:

- introduction of noise

$\mathcal{K} = \{ JournalPaper \sqsubseteq Paper,\ ConferencePaper \sqsubseteq Paper,\ ConferencePaper(a), Author(a)\ \}$
$\mathcal{K}$ is Consistent !!!
**Cause** Axiom: $Author \equiv \neg ConferencePaper$ **missing**

- counterintuitive inferences

$\mathcal{K} = \{ JournalPaper \sqsubseteq Paper,\ ConferencePaper \sqsubseteq Paper,\ ConferencePaper(a)\ \}$

$\mathcal{K} \models JournalPaper(a)$?
Answer: Unknown
**Cause** Axiom: $JournalPaper \equiv \neg ConferencePaper$ **missing**

- hard collecting negative examples when adopting numeric approaches

**Observation:** extensions of disjoint concepts do not overlap

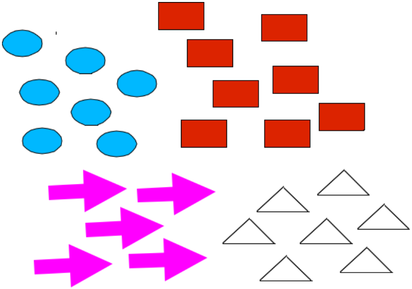
**Question:** would it be possible to *automatically capture* disjointness axioms by analyzing the data configuration/distribution?

**Idea:** Exploiting **(Conceptual) clustering methods** for the purpose

# Clustering Methods

Unsupervised inductive learning methods that organize a collection of unlabeled resources into meaningful clusters such that

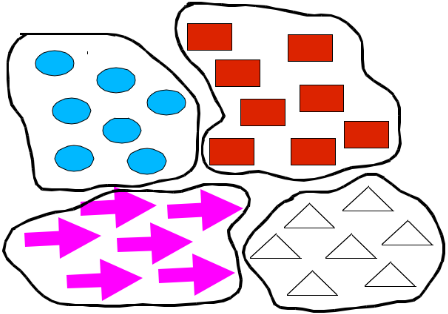
- intra-cluster *similarity* is high
- inter-cluster *similarity* is low



# Clustering Methods

Unsupervised inductive learning methods that organize a collection of unlabeled resources into meaningful clusters such that

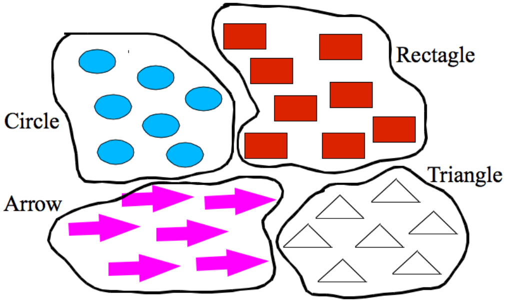
- intra-cluster *similarity* is high
- inter-cluster *similarity* is low



# Clustering Methods

Unsupervised inductive learning methods that organize a collection of unlabeled resources into meaningful clusters such that

- intra-cluster *similarity* is high
- inter-cluster *similarity* is low



**Observation:** extensions of disjoint concepts do not overlap

**Question:** would it be possible to *automatically capture* them by analyzing the data configuration/distribution?

**Idea:** Exploiting **(Conceptual) clustering methods** for the purpose

### Definition (Problem Definition)

Given

- a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$
- a set of individuals (aka entities) $\mathbf{I} \subseteq \mathsf{Ind}(\mathcal{A})(\mathcal{A})$

Find

- $n$ pairwise disjoint clusters $\{\mathbf{C}_1, \ldots, \mathbf{C}_n\}$
- for each $i = 1, \ldots, n$, a concept description $D_i$ that describes $\mathbf{C}_i$, such that:
  - $\forall a \in \mathbf{C}_i : \ \mathcal{K} \models D_i(a)$
  - $\forall b \in \mathbf{C}_j, j \neq i : \ \mathcal{K} \models \neg D_i(b)$.
- Hence $\forall D_i, D_j, i \neq j : \ \mathcal{K} \models D_j \sqsubseteq \neg D_i$.

# Learning Disjointness Axioms: Developed Methods

**Statistical-based approach**

- NAR - exploiting negative association rules [Fleischhacker and Völker, 2011]
- PCC - exploiting Pearson's correlation coeff. [Völker *et al.*, 2015]

do not exploit any background knowledge and reasoning capabilities

# Terminological Cluster Tree

Defined a method [Rizzo *et al.*, 2021] for eliciting disjointness axioms[4]

- solving a clustering problem via learning Terminological Cluster Trees
- providing a concept description for each cluster

**Definition (Terminological cluster tree (TCT))**

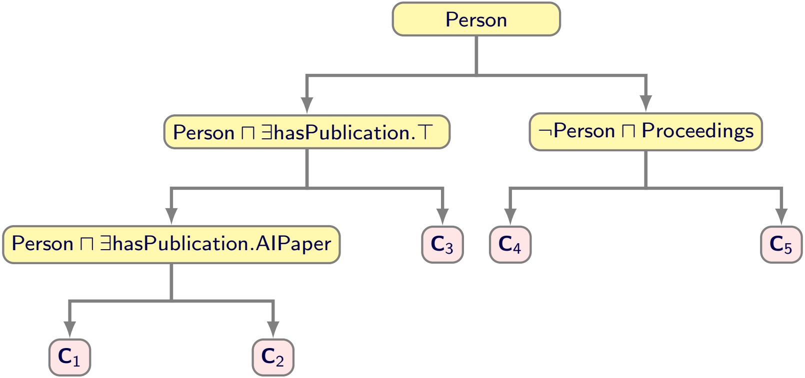
A binary logical tree where

- a leaf node stands for a cluster of individuals $\mathbf{C}$
- each inner node contains a description $D$ (over the signature of $\mathcal{K}$)
- each departing edge corresponds to positive (left) and negative (right) examples of $D$

[4] Implemented system publicly available at `https://github.com/Giuseppe-Rizzo/TCTnew`

# Example of TCT

Given $\mathbf{I} \subseteq \mathrm{Ind}(\mathcal{A})(\mathcal{A})$, an example of TCT describing the AI research community



# Collecting Disjointness Axioms

Given a TCT **T**:

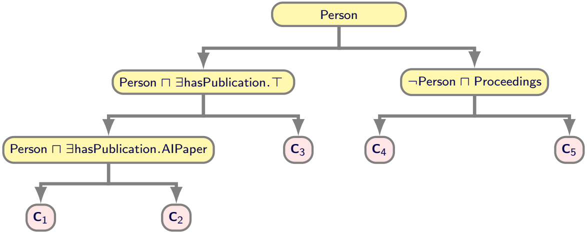
Step I:

- Traverse the **T** to collect the concept descriptions describing the clusters at the leaves
- A set of concepts **CS** is obtained

Step II:

- A set of candidate axioms **A** is generated from **CS**:
  - an axiom $D \sqsubseteq \neg E$ ($D, E \in \mathbf{CS}$) is generated if
    - $D \not\equiv E$ (or $D \not\sqsubseteq E$ or viceversa - *reasoner needed*)
    - $E \sqsubseteq \neg D$ has not been generated

# Collecting Disjointness Axioms: Example




$$
\begin{aligned}
CS = \{ \quad & \text{Person}, \\
& \text{Person} \sqcap \exists \text{hasPublication}.\top, \\
& \neg(\text{Person} \sqcap \exists \text{hasPublication}.\top) \\
& \text{Person} \sqcap \exists \text{hasPublication}.\text{AIPaper} \\
& \neg\text{Person} \sqcap \text{Proceedings} \cdots \}
\end{aligned}
$$

Axiom1: $Person \sqcap \exists hasPublication.AIPaper \sqsubseteq \neg(\neg Person \sqcap Proceedings)$

Axiom2: $\cdots$

# Inducing a TCT

Given the set of individuals $\mathbf{I}$ and $\top$ concept

*Divide-and-conquere* approach adopted

- **Base Case:** test the $\textsc{stopCondition}$
  - the cohesion of the cluster $\mathbf{I}$ exceeds a threshold $\nu$
    - distance between *medoids* below a threshold $\nu$
- **Recursive Step** ($\textsc{stopCondition}$ does not hold):
  - a set $\mathbf{S}$ of refinements of the current (parent) description $C$ generated
  - the $\textsc{bestConcept}$ $E^* \in \mathbf{S}$ is selected and installed as *current node*
    - the one showing the *best cluster separation* $\Leftrightarrow$ with max distance between the *medoids* of its positive $P$ and negative $N$ individuals
  - $\mathbf{I}$ is $\textsc{split}$ in:
    - $I_{left} \subseteq \mathbf{I} \leftrightarrow$ individuals with the smallest distance wrt the *medoid* of $P$
    - $I_{right} \subseteq \mathbf{I} \leftrightarrow$ individuals with the smallest distance wrt the *medoid* of $N$
    - *reasoner employed* for collecting $P$ and $N$

**Note:** *Number of clusters not required* - obtained from data distribution

# Lesson Learnt from experiments I

## Experiments performed on ontologies publicly available

- Goal I: Re-discover a target axiom (existing in $\mathcal{K}$)
  - Setting:
    - A copy of each ontology is created removing a target axiom
    - Threshold $\nu = 0.9, 0.8, 0.7$
    - Metrics # discovered axioms and #cases of inconsistency
  - Results:
    - target axioms rediscovered for almost all cases
    - *additional* disjointness *axioms discovered* in a significant number
    - limited number of inconsistencies found

| *Ontology* | TCT 0.9 | | TCT 0.8 | | TCT 0.7 | |
|---|---|---|---|---|---|---|
| | #inc. | #ax's | #inc. | #ax's | #inc. | #ax's |
| BioPax | 2 | 53 | 2 | 53 | 3 | 52 |
| NTN | 10 | 70 | 9 | 73 | 10 | 75 |
| Financial | 0 | 125 | 0 | 126 | 0 | 127 |
| GeoSkills | 2 | 345 | 1 | 347 | 4 | 347 |
| Monetary | 0 | 432 | 0 | 432 | 0 | 433 |
| DBPedia3.9 | 45 | 45 | 44 | 44 | 43 | 43 |

# Lesson Learnt from experiments II

Goal II:

- Re-discover randomly selected target axioms added according to the **Strong Disjointness Assumption** [Schlobach, 2005]
  - two sibling concepts in a subsumption hierarchy considered as disjoint
- comparative analysis with statistical-based methods [Völker *et al.*, 2015; Fleischhacker and Völker, 2011]
  - PCC - based on *Pearson's correlation coefficient*
  - NAR - exploiting *negative association rules*
- Setting:
  - A copy of each ontology created removing 20%, 50%, 70% of the disjointness axioms
    - The copy used to induce TCT - $\nu = 0.9, 0.8, 0.7$ - # Run: 10 times
  - **Metrics**: rate of **rediscovered** target axioms, #cases of inconsistency, # addional discovered axioms

# Lesson Learnt from experiments III

- Results:
  - *almost all axioms rediscovered*
    - Rate decreases when larger fractions of axioms removed, *as expected*
  - *TCT outperforms PCC and NAR* wrt *additionally discovered axioms* whilst introducing limited inconsistency
    - TCT allows to express complex disjointness axioms
    - PCC and NAR tackle only disjointness between concept names

**Exploiting the $\mathcal{K}$ as well as the data distribution** improves **disjointness axioms discovery**

# Example of axioms

## Successfully discovered axioms

- ExternalReferenceUtilityClass $\sqcap \exists$TAXONREF.$\top$
  disjoint with
  xref
- Activity
  disjoint with
  Person $\sqcap \exists$nationality.United_states
- Person $\sqcap$ hasSex.Male ($\equiv$ Man)
  disjoint with
  SupernaturalBeing $\sqcap$ God ($\equiv$ God)

## Not discovered axioms

- Actor disjoint with Artefact

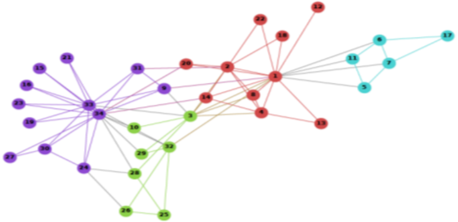
(concepts with few instances)

# Numeric-based Methods for Knowledge Graph Refinement

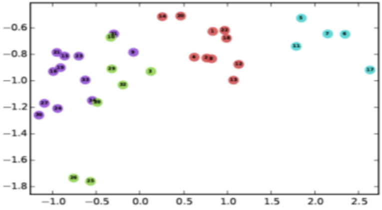
# KG Embedding Models...

*Vector embedding models* largely investigated [Cai *et al.*, 2018]

- convert data graph into an optimal low-dimensional space
- *Graph structural information* preserved as much as possible
- CWA (or LCWA) mostly adopted vs. OWA
- *schema level information* and *reasoning* capabilities almost disregarded



Input




Output [5]

[5] Picture from https://laptrinhx.com/node2vec-graph-embedding-method-2620064815/

# ...KG Embedding Models...

**Graph embedding methods differ in their main building blocks:** [Ji *et al.*, 2021]

the representation space: point-wise, complex, discrete, Gaussian, manifold, etc.

the encoding model: linear, factorization, neural models, etc.

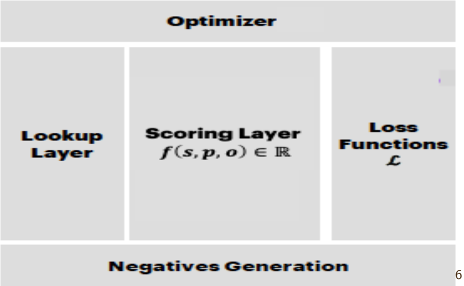
the scoring function: based on distance, energy, semantic matching, other criteria, etc.

# ...KG Embedding Models

**Goal**

Learning embeddings s.t.

- score of a valid (positive) triple is higher than
- the score of an invalid (negative) triple



[6]

[6] Picture from "ECAI-20 Tutorial: Knowledge Graph Embeddings: From Theory to Practice"

**Idea:** Enhance KGE through Background Knowledge Injection

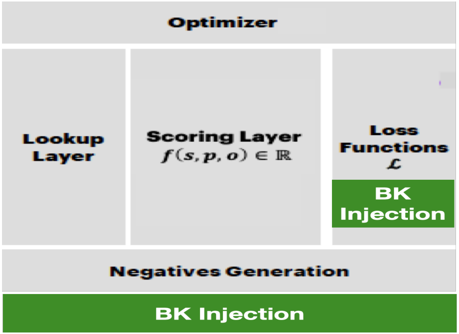
By two components:

**Reasoning:** used for generating negative triples

**Axioms:** `domain`, `range`, `disjointWith`, `functionalProperty`;

**BK Injection:** defines constraints on functions, corresponding to the considered axioms, *guiding the way embedding are learned*

**Axioms:** `equivClass`, `equivProperty`, `inverseOf` and `subClassOf`.



# Other KG Embedding Methods Leveraging BK

- Jointly embedding KGs and logical rules [Guo *et al.*, 2016]
  - triples represented as atomic formulae
  - rules represented as complex formulae modeled by t-norm fuzzy logics
- Adversarial training exploiting Datalog clauses encoding assumptions to regularize neural link predictors [Minervini *et al.*, 2017a]

A specific form of BK required, not directly applicable to KGs

## An approach to learn embeddings exploiting BK

[d'Amato *et al.*, 2021]

TRANSOWL

TransE

TRANSROWL TRANSROWL$^R$

TransR

Could be applied to more complex KG embedding methods with additional formalization

# TransOWL...

## TransOWL maintains TransE setting

TransE [Bordes *et al.*, 2013] learns the vector embedding by minimizing

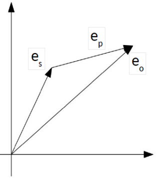
*Margin-based loss function*

$$L = \sum_{\substack{\langle s,p,o\rangle \in \Delta \\ \langle s',p,o'\rangle \in \Delta'}} \left[\gamma + f_p(\mathbf{e}_s, \mathbf{e}_o) - f_p(\mathbf{e}_{s'}, \mathbf{e}_{o'})\right]_+$$

where $[x]_+ = \max\{0, x\}$, and $\gamma \geq 0$

*Score function*

similarity (negative $L_1$ or $L_2$ distance) of the translated subject embedding $(\mathbf{e}_s + \mathbf{e}_p)$ to the object embedding $\mathbf{e}_o$:




$$f_p(\mathbf{e}_s, \mathbf{e}_o) = -\|(\mathbf{e}_s + \mathbf{e}_p) - \mathbf{e}_o\|_{\{1,2\}}.$$

# ...TransOWL

- Derive *further triples to be considered for training* via schema axioms
  - `equivClass`, `equivProperty`, `inverseOf` and `subClassOf`
- More complex loss function
  - adding a number of terms consistently with the constraints

$$
\begin{aligned}
L \;=\; & \overbrace{\sum_{\substack{\langle h,r,t\rangle \in \Delta \\ \langle h',r,t'\rangle \in \Delta'}} [\gamma + f_r(h,t) - f_r(h',t')]_+}^{\textsc{TransE}\ \textit{loss function}} + \sum_{\substack{\langle t,q,h\rangle \in \Delta_{\text{inverseOf}} \\ \langle t',q,h'\rangle \in \Delta'_{\text{inverseOf}}}} [\gamma + f_q(t,h) - f_q(t',h')]_+ \\
&+ \sum_{\substack{\langle h,s,t\rangle \in \Delta_{\text{equivProperty}} \\ \langle h',s,t'\rangle \in \Delta'_{\text{equivProperty}}}} [\gamma + f_s(h,t) - f_s(h',t')]_+ + \sum_{\substack{\langle h,\text{typeOf},l\rangle \in \Delta \cup \in \Delta_{\text{equivClass}} \\ \langle h',\text{typeOf},l'\rangle \in \Delta' \cup \Delta'_{\text{equivClass}}}} [\gamma + f_{\text{typeOf}}(h,l) - f_{\text{typeOf}}(h',l')]_+ \\
&+ \sum_{\substack{\langle h,\text{subClassOf},p\rangle \in \Delta_{\text{subClass}} \\ \langle h',\text{subClassOf},p'\rangle \in \Delta'_{\text{subClass}}}} [(\gamma - \beta) + f(h,p) - f(h',p')]_+
\end{aligned}
$$

where $q \equiv r^-$, $s \equiv r$ (properties), $l \equiv t$ and $t \sqsubseteq p$ (classes) and $f(h,p) = \|\mathbf{e}_h - \mathbf{e}_p\|$

# TRANSROWL...

TRANSROWL

- adopts the same approach of TRANSOWL
- *is derived from* TRANSR [Lin *et al.*, 2015]

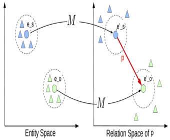
TRANSE ⇒ poor modeling *reflexive* and *non* 1-to-1 relations (e.g. typeOf)

TRANSR ⇒ more suitable to handle such specificity

TRANSR adopts TRANSE *loss function*

*Score function*

preliminarily projects $\mathbf{e}_s$ and $\mathbf{e}_o$ to the different $d$-dimensional space of the relational embeddings $\mathbf{e}_p$ through a suitable matrix $\mathbf{M} \in \mathbb{R}^{k \times d}$:




$$f'_p(\mathbf{e}_s, \mathbf{e}_o) = -\|(\mathbf{M}\mathbf{e}_s + \mathbf{e}_p) - \mathbf{M}\mathbf{e}_o\|_{\{1,2\}}.$$

where $\mathbf{e}'_s = \mathbf{M}\mathbf{e}_s$ and $\mathbf{e}'_o = \mathbf{M}\mathbf{e}_o$

# ...TRANSROWL

- TRANSOWL loss function adopted plus weighting parameters
  - equivClass, equivProperty, inverseOf and subClassOf
- TRANSR score function adopted

$$
\begin{aligned}
L \quad = & \sum_{\substack{\langle h,r,t\rangle \in \Delta \\ \langle h',r,t'\rangle \in \Delta'}} [\gamma + f'_r(h,t) - f'_r(h',t')]_+ + \lambda_1 \sum_{\substack{\langle t,q,h\rangle \in \Delta_{\text{inverseOf}} \\ \langle t',q,h'\rangle \in \Delta_{\text{inverseOf}'}}} [\gamma + f'_q(t,h) - f'_q(t',h')]_+ \\
& + \lambda_2 \sum_{\substack{\langle h,s,t\rangle \in \Delta_{\text{equivProperty}} \\ \langle h',s,t'\rangle \in \Delta_{\text{equivProperty}'}}} [\gamma + f'_s(h,t) - f'_s(h',t')]_+ + \lambda_3 \sum_{\substack{\langle h,\text{typeOf},l\rangle \in \Delta \cup \Delta_{\text{equivClass}} \\ \langle h',\text{typeOf},l'\rangle \in \Delta' \cup \Delta'_{\text{equivClass}}}} [\gamma + f'_{\text{typeOf}}(h,l) - f'_{\text{typeOf}}(h',l')]_+ \\
& + \lambda_4 \sum_{\substack{\langle t,\text{subClassOf},p\rangle \in \Delta_{\text{subClass}} \\ \langle t',\text{subClassOf},p'\rangle \in \Delta_{\text{subClass}'}}} [(\gamma - \beta) + f'(t,p) - f'(t',p')]_+
\end{aligned}
$$

where

- $q \equiv r^-$, $s \equiv r$ (properties), $l \equiv t$ and $t \sqsubseteq p$ (classes)
- the parameters $\lambda_i$, $i \in \{1, \dots, 4\}$, weigh the influence that each function term has during the learning phase

# TransROWL$^R$...

TransROWL$^R$ adopts axiom-based regularization of *the loss function*, as for TransE$^R$ [Minervini *et al.*, 2017b]

- by adding specific constraints to the loss function rather than
- explicitly derive additional triples during training

TransE$^R$ adopt TransE *score* and *loss function*
adds to the loss function *axiom-based regularizers* for inverse and equivalent property constraints

*Loss function*

$$L = \sum_{\substack{\langle h,r,t\rangle \in \Delta \\ (h',r',t') \in \Delta'}} [\gamma + f_r(h,t) - f_r(h',t')]_+ + \lambda \sum_{r \equiv q^- \in \mathcal{T}_{\text{inverseOf}}} \|r + q\| + \lambda \sum_{r \equiv p \in \mathcal{T}_{\text{equivProp}}} \|r - p\|$$

where $\mathcal{T}_{\text{inverseOf}}$ $\mathcal{T}_{\text{equivProp}}$ set of inverse properties and equivalent properties

# ...TransROWL$^R$

- TransR score function adopted
- *additional regularizers needed* for `equivalentClass` and `subClassOf` axioms
- *further constraints on the projection matrices* associated to relations

*Loss function*

$$
\begin{aligned}
L \quad = \quad & \sum_{\substack{\langle h,r,t\rangle \in \Delta \\ \langle h',r',t'\rangle \in \Delta'}} [\gamma + f'_r(h,t) - f'_r(h',t')]_+ \\
& + \lambda_1 \sum_{r \equiv q^- \in \mathcal{T}_{\text{inverseOf}}} \|r + q\| \; + \lambda_2 \sum_{r \equiv q^- \in \mathcal{T}_{\text{inverseOf}}} \|M_r - M_q\| \\
& + \lambda_3 \sum_{r \equiv p \in \mathcal{T}_{\text{equivProp}}} \|r - p\| \; + \lambda_4 \sum_{r \equiv p \in \mathcal{T}_{\text{equivProp}}} \|M_r - M_p\| \\
& + \lambda_5 \sum_{e' \equiv e'' \in \mathcal{T}_{\text{equivClass}}} \|e' - e''\| \; + \lambda_6 \sum_{s' \subseteq s'' \in \mathcal{T}_{\text{subClass}}} \|1 - \beta - (s' - s'')\|
\end{aligned}
$$

Additional term for projection matrices required for `inverseOf` and `equivProp` triples to favor the equality of their projection matrices

# Lesson Learnt from Experiments...

**Goal: Assessing the benefit of exploiting BK**

- Comparing[7] TransOWL, TransROWL, TransROWL$^R$ over to the original models TransE and TransR as a baseline

Perfomances tested on:

- Link Prediction task
- Triple Classification task
- Standard metrics adopted

KGs adopted:

| *KG* | *#Triples* | *#Entities* | *#Relationships* |
|---|---|---|---|
| DBpedia15K | 180000 | 12800 | 278 |
| DBpedia100K | 600000 | 100000 | 321 |
| DBpediaYAGO | 290000 | 88000 | 316 |
| NELL[8] | 150000 | 68000 | 272 |

[7] All methods implemented as publicly available systems https://github.com/Keehl-Mihael/TransROWL-HRS

[8] equivalentClass and equivalentProperty missing; limited number of typeOf-triples; abundance of subClassOf-triples

# ...Lesson Learnt from Experiments

- Best performance achieved by TransROWL, in most of the cases, and TransROWL$^R$
- TransROWL slightly superior performance of TransROWL$^R$

As for NELL, the models showed oscillating performances wrt the baselines

- NELL was aimed at testing in condition of larger incompleteness
  - `equivalentClass` and `equivalentProperty` **missing**
  - low number of typeOf-triples per entity

# Conclusions

# Conclusions

**Machine Learning methods**

- could be usefully exploited for ontology mining and KG refinement
- suitable also in case of incoherent/noisy KBs
- **can be seen as an additional layer on top of deductive reasoning** for *new/additional forms of approximated reasoning capabilities*

**Adopting ML solutions could be simple in principle**

- often instantiating an existing learning schema is just needed
- *Alert*
  - understand the meaning of each component for instantiating a learning schema correctly
  - it could be the case that some components require newly developed solutions
    - e.g. new similarity measure for expressive representations, suitable refinement operators, injecting BK

That's all!

Questions ?

**Claudia d'Amato**
Computer Science Department
University of Bari, Bari - Italy
email:
`claudia.damato@uniba.it`



# References I


Berners-Lee, T. (2006). Linked data - design issues.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, **284**(5), 34–43.

Bloehdorn, S. and Sure, Y. (2007). Kernel methods for mining instance data in ontologies. In K. Aberer, K. Choi, N. F. Noy, D. Allemang, K. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 58–71. Springer.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In C. J. C. Burges *et al.*, editors, *Proceedings of NIPS 2013*, pages 2787–2795. Curran Associates, Inc.

Cai, H., Zheng, V. W., and Chang, K. C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.*, **30**(9), 1616–1637.

d'Amato, C. (2007). Similarity-based learning methods for the semantic web. `http://www.di.uniba.it/~cdamato/PhDThesis_dAmato.pdf`. PhD Thesis.

d'Amato, C. (2020). Machine learning for the semantic web: Lessons learnt and next research directions. *Semantic Web*, **11**(1), 195–203.

# References II


d'Amato, C., Staab, S., and Fanizzi, N. (2008a). On the influence of description logics ontologies on conceptual similarity. In A. Gangemi and J. Euzenat, editors, *Knowledge Engineering: Practice and Patterns, 16th International Conference, EKAW 2008, Acitrezza, Italy, September 29 - October 2, 2008. Proceedings*, volume 5268 of *Lecture Notes in Computer Science*, pages 48–63. Springer.

d'Amato, C., Fanizzi, N., and Esposito, F. (2008b). Query answering and ontology population: An inductive approach. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings*, volume 5021 of *Lecture Notes in Computer Science*, pages 288–302. Springer.

d'Amato, C., Fanizzi, N., and Esposito, F. (2010). Inductive learning for the semantic web: What does it buy? *Semantic Web*, **1**(1-2), 53–59.

d'Amato, C., Quatraro, N. F., and Fanizzi, N. (2021). Injecting background knowledge into embedding models for predictive tasks on knowledge graphs. In R. Verborgh, K. Hose, H. Paulheim, P. Champin, M. Maleshkova, Ó. Corcho, P. Ristoski, and M. Alam, editors, *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*, volume 12731 of *Lecture Notes in Computer Science*, pages 441–457. Springer.

De Raedt, L. (2008). *Logical and Relational Learning: From ILP to MRDM (Cognitive Technologies)*. Springer-Verlag, Berlin, Heidelberg.

Fanizzi, N. (2011). Concept induction in Description Logics using information-theoretic heuristics. *Int. J. Semantic Web Inf. Syst.*, **7**(2), 23–44.

# References III


Fanizzi, N. and d'Amato, C. (2006). A declarative kernel for $\mathcal{ALC}$ concept descriptions. In F. Esposito, Z. W. Ras, D. Malerba, and G. Semeraro, editors, *Foundations of Intelligent Systems, 16th International Symposium, ISMIS 2006, Bari, Italy, September 27-29, 2006, Proceedings*, volume 4203 of *Lecture Notes in Computer Science*, pages 322–331. Springer.

Fanizzi, N., d'Amato, C., and Esposito, F. (2008). DL-Foil. Concept learning in Description Logics. In F. Zelezný and N. Lavrač, editors, *Proceedings of ILP2008*, volume 5194 of *LNAI*, pages 107–121. Springer.

Fanizzi, N., d'Amato, C., and Esposito, F. (2010). Induction of concepts in web ontologies through terminological decision trees. In J. L. Balcázar *et al.*, editors, *Proceedings of ECML/PKDD 2010, Part I*, volume 6321 of *LNAI*, pages 442–457. Springer.

Fanizzi, N., d'Amato, C., and Esposito, F. (2012a). Induction of robust classifiers for web ontologies through kernel machines. *J. Web Sem.*, **11**, 1–13.

Fanizzi, N., d'Amato, C., Esposito, F., and Minervini, P. (2012b). Numeric prediction on OWL knowledge bases through terminological regression trees. *Int. J. Semantic Comput.*, **6**(4), 429–446.

Fanizzi, N., Rizzo, G., d'Amato, C., and Esposito, F. (2018). Dlfoil: Class expression learning revisited. In C. Faron-Zucker, C. Ghidini, A. Napoli, and Y. Toussaint, editors, *Knowledge Engineering and Knowledge Management - 21st International Conference, EKAW 2018, Nancy, France, November 12-16, 2018, Proceedings*, volume 11313 of *Lecture Notes in Computer Science*, pages 98–113. Springer.

# References IV


Fanizzi, N., Rizzo, G., and d'Amato, C. (2019). Boosting DL concept learners. In P. Hitzler, M. Fernández, K. Janowicz, A. Zaveri, A. J. G. Gray, V. López, A. Haller, and K. Hammar, editors, *The Semantic Web - 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2-6, 2019, Proceedings*, volume 11503 of *Lecture Notes in Computer Science*, pages 68–83. Springer.

Flach, P. (2012). *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge University Press, New York, NY, USA.

Fleischhacker, D. and Völker, J. (2011). Inductive learning of disjointness axioms. In R. Meersman and et. al., editors, *On the Move to Meaningful Internet Systems: OTM 2011 - Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2011, Proceedings, Part II*, volume 7045 of *Lecture Notes in Computer Science*, pages 680–697. Springer.

Guo, S., Wang, Q., Wang, L., Wang, B., and Guo, L. (2016). Jointly embedding knowledge graphs and logical rules. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 192–202, Austin, Texas. Association for Computational Linguistics.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutiérrez, C., Gayo, J. L., Kirrane, S., Neumaier, S., Polleres, A., Navigli, R., Ngomo, A. N., Rashid, S., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., and Zimmermann, A. (2021). Knowledge graphs. *ACM Computing Surveys*, **54**, 1–37.

# References V

Huang, Y., Tresp, V., Bundschus, M., Rettinger, A., and Kriegel, H. (2010). Multivariate prediction for learning on the semantic web. In P. Frasconi and F. A. Lisi, editors, *Inductive Logic Programming - 20th International Conference, ILP 2010, Florence, Italy, June 27-30, 2010. Revised Papers*, volume 6489 of *Lecture Notes in Computer Science*, pages 92–104. Springer.

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

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. (2021). A survey on knowledge graphs: representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Lehmann, J. and Hitzler, P. (2010). Concept learning in description logics using refinement operators. *Mach. Learn.*, **78**(1-2), 203–250.

Lehmann, J., Auer, S., Bühmann, L., and Tramp, S. (2011). Class expression learning for ontology engineering. *Journal of Web Semantics*, **9**, 71 – 81.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *AAAI 2015 Proceedings*, pages 2181–2187. AAAI Press.

Luger, G. F. (2005). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Addison Wesley, 5 edition.

MacKay, D. J. C. (2002). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA.

# References VI


Melo, A. and Paulheim, H. (2019). Local and global feature selection for multilabel classification with binary relevance - an empirical comparison on flat and hierarchical problems. *Artif. Intell. Rev.*, **51**(1), 33–60.

Melo, A., Paulheim, H., and Völker, J. (2016). Type prediction in RDF knowledge bases using hierarchical multilabel classification. In R. Akerkar, M. Plantié, S. Ranwez, S. Harispe, A. Laurent, P. Bellot, J. Montmain, and F. Trousset, editors, *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics, WIMS 2016, Nîmes, France, June 13-15, 2016*, pages 14:1–14:10. ACM.

Melo, A., Völker, J., and Paulheim, H. (2017). Type prediction in noisy RDF knowledge bases using hierarchical multilabel classification with graph and latent features. *Int. J. Artif. Intell. Tools*, **26**(2), 1760011:1–1760011:32.

Minervini, P., Fanizzi, N., d'Amato, C., and Esposito, F. (2015). Scalable learning of entity and predicate embeddings for knowledge graph completion. In T. Li, L. A. Kurgan, V. Palade, R. Goebel, A. Holzinger, K. Verspoor, and M. A. Wani, editors, *14th IEEE International Conference on Machine Learning and Applications, ICMLA 2015, Miami, FL, USA, December 9-11, 2015*, pages 162–167. IEEE.

Minervini, P., Demeester, T., Rocktäschel, T., and Riedel, S. (2017a). Adversarial sets for regularising neural link predictors. In G. Elidan *et al.*, editors, *UAI 2017 Proceedings*. AUAI Press.

# References VII


Minervini, P., Costabello, L., Muñoz, E., Novácek, V., and Vandenbussche, P. (2017b). Regularizing knowledge graph embeddings via equivalence and inversion axioms. In M. Ceci *et al.*, editors, *Proceedings of ECML PKDD 2017, Part I*, volume 10534 of *LNAI*, pages 668–683. Springer.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Peixoto, R., Hassan, T., Cruz, C., Bertaux, A., and Silva, N. (2016). Hierarchical multi-label classification using web reasoning for large datasets. *Open J. Semantic Web*, **3**(1), 1–15.

Rizzo, G., d'Amato, C., Fanizzi, N., and Esposito, F. (2015a). Inductive classification through evidence-based models and their ensembles. In F. Gandon, M. Sabou, H. Sack, C. d'Amato, P. Cudré-Mauroux, and A. Zimmermann, editors, *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings*, volume 9088 of *Lecture Notes in Computer Science*, pages 418–433. Springer.

Rizzo, G., d'Amato, C., and Fanizzi, N. (2015b). On the effectiveness of evidence-based terminological decision trees. In F. Esposito, O. Pivert, M. Hacid, Z. W. Ras, and S. Ferilli, editors, *Foundations of Intelligent Systems - 22nd International Symposium, ISMIS 2015, Lyon, France, October 21-23, 2015, Proceedings*, volume 9384 of *Lecture Notes in Computer Science*, pages 139–149. Springer.

Rizzo, G., d'Amato, C., Fanizzi, N., and Esposito, F. (2016). Approximating numeric role fillers via predictive clustering trees for knowledge base enrichment in the web of data. In T. Calders, M. Ceci, and D. Malerba, editors, *Discovery Science - 19th International Conference, DS 2016, Bari, Italy, October 19-21, 2016, Proceedings*, volume 9956 of *Lecture Notes in Computer Science*, pages 101–117.

Rizzo, G., d'Amato, C., Fanizzi, N., and Esposito, F. (2017a). Terminological cluster trees for disjointness axiom discovery. In E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, and O. Hartig, editors, *The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I*, volume 10249 of *Lecture Notes in Computer Science*, pages 184–201.

Rizzo, G., d'Amato, C., Fanizzi, N., and Esposito, F. (2017b). Tree-based models for inductive classification on the web of data. *J. Web Sem.*, **45**, 1–22.

Rizzo, G., Fanizzi, N., d'Amato, C., and Esposito, F. (2018). Approximate classification with web ontologies through evidential terminological trees and forests. *Int. J. Approx. Reasoning*, **92**, 340–362.

Rizzo, G., Fanizzi, N., and d'Amato, C. (2020). Class expression induction as concept space exploration: From dl-foil to dl-focl. *Future Gener. Comput. Syst.*, **108**, 256–272.

Rizzo, G., d'Amato, C., and Fanizzi, N. (2021). An unsupervised approach to disjointness learning based on terminological cluster trees. *Semantic Web Journal*, **12**(3), 423–447.

# References IX


Schlobach, S. (2005). Debugging and semantic clarification by pinpointing. In A. Gómez-Pérez and J. Euzenat, editors, *The Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29 - June 1, 2005, Proceedings*, volume 3532 of *Lecture Notes in Computer Science*, pages 226–240. Springer.

Shadbolt, N., Hall, W., and Berners-Lee, T. (2006). The semantic web revisited. *Intelligent Systems, IEEE*, **21**(3), 96–101.

Tran, A. C., Dietrich, J., Guesgen, H. W., and Marsland, S. (2012). An approach to parallel class expression learning. In A. Bikakis and A. Giurca, editors, *Proceedings of RuleML 2012*, volume 7438 of *LNCS*, pages 302–316. Springer.

Tran, A. C., Dietrich, J., Guesgen, H. W., and Marsland, S. (2017). Parallel symmetric class expression learning. *Journal of Machine Learning Research*, **18**, 64:1–64:34.

Völker, J., Fleischhacker, D., and Stuckenschmidt, H. (2015). Automatic acquisition of class disjointness. *Journal of Web Semantics*, **35**, 124–139.