Ontology Learning

(with emphasis on refinement operator based learning)

Jens Lehmann, Johanna Völker

UNIVERSITÄT LEIPZIG

UNIVERSITÄT MANNHEIM

2010-09-02

Outline

Outline

Definition: Ontology Learning

- *"Ontology Learning is a subtask of information extraction. The goal of ontology learning is to (semi-)automatically extract relevant concepts and relations from a given corpus or other kinds of data sets to form an ontology."* (Wikipedia, today)

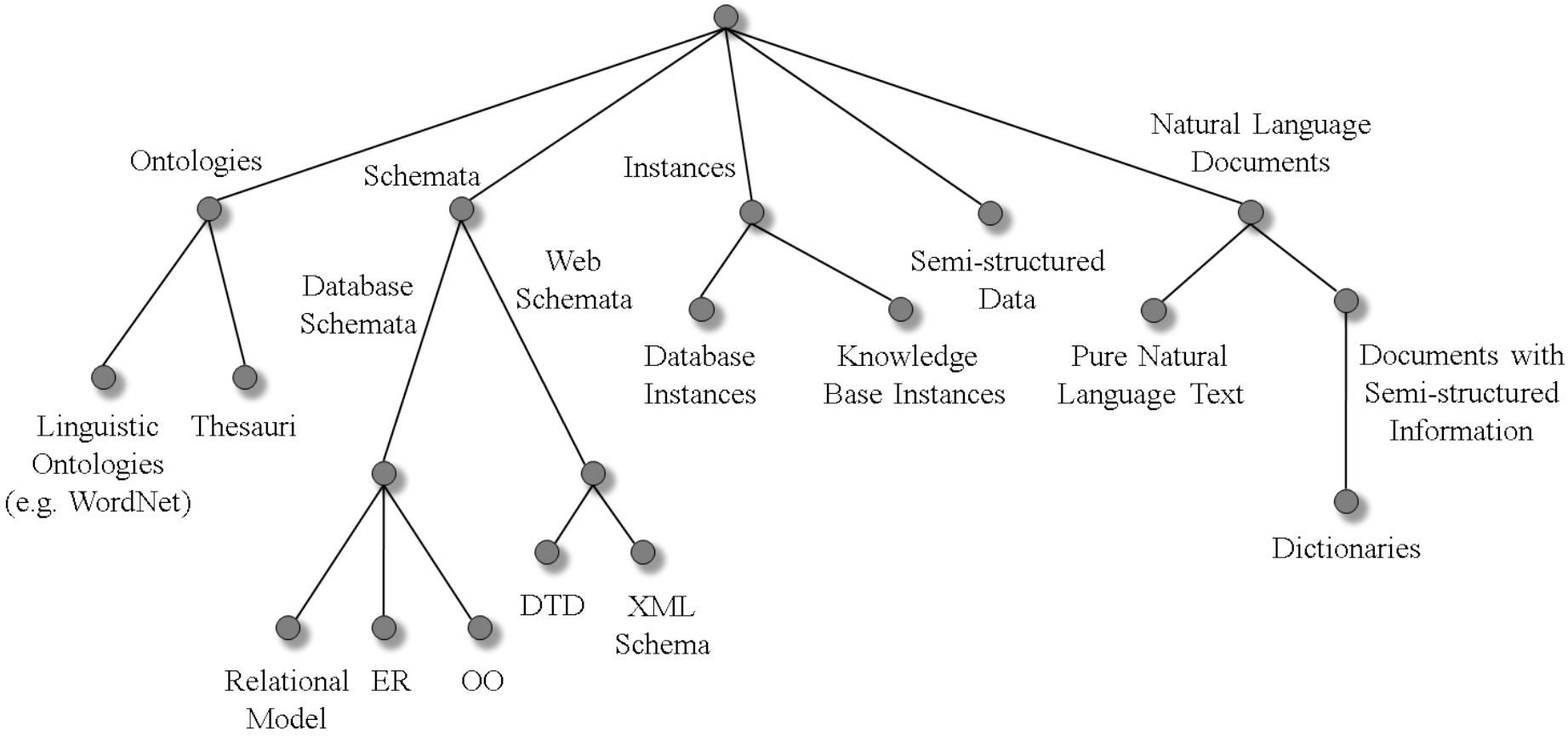
Definition: Ontology Learning

- *"Ontology Learning is a subtask of information extraction. The goal of ontology learning is to (semi-)automatically extract relevant concepts and relations from a given corpus or other kinds of data sets to form an ontology."* (Wikipedia, today)
- *"Ontology Learning is a mechanism for semi-automatically supporting the ontology engineer in engineering ontologies."*
  A. D. Mädche. Ontology Learning for the Semantic Web. Dissertation. Universität Karlsruhe, 2001

Definition: Ontology Learning

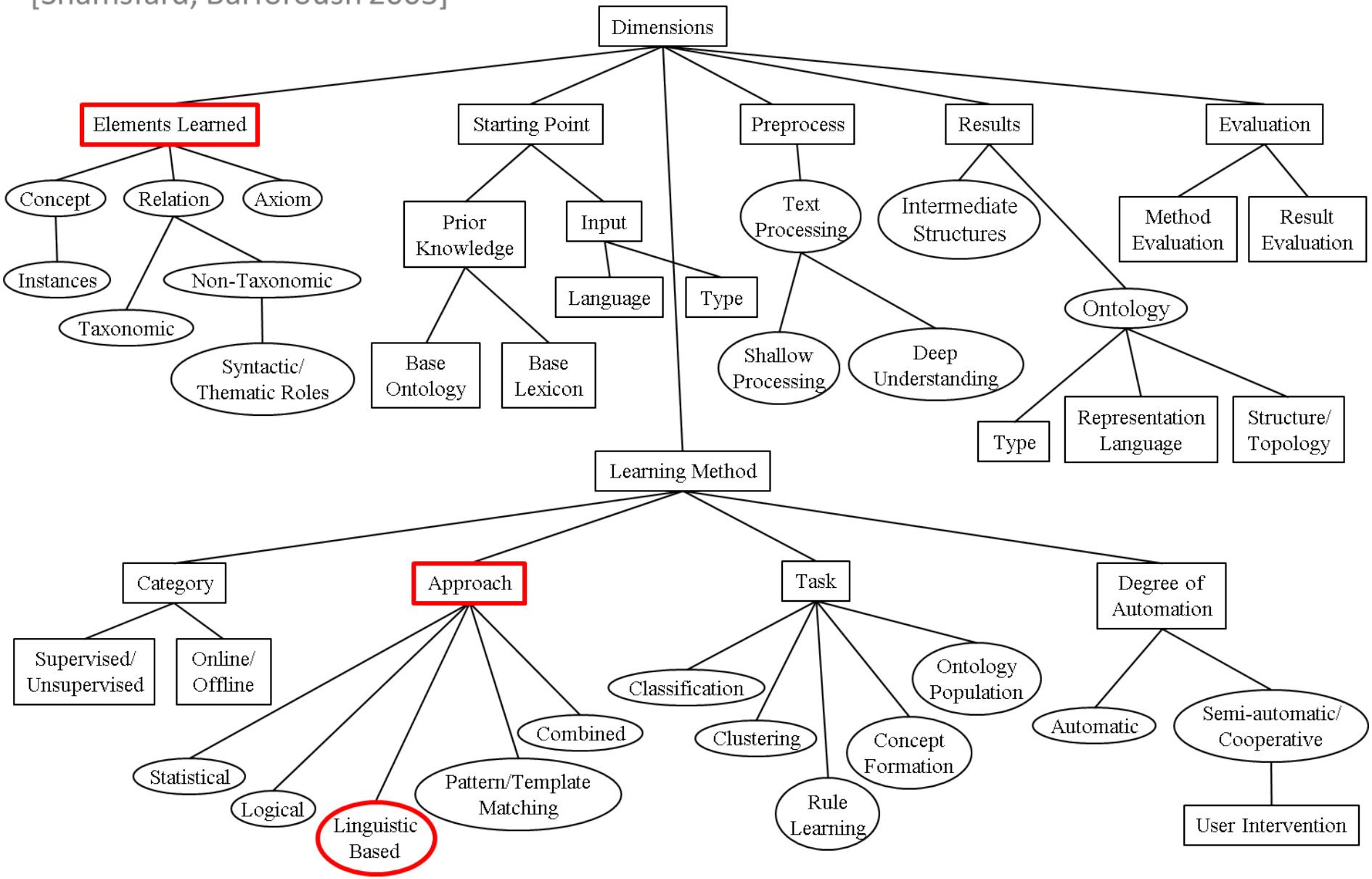
- *"Ontology Learning is a subtask of information extraction. The goal of ontology learning is to (semi-)automatically extract relevant concepts and relations from a given corpus or other kinds of data sets to form an ontology."* (Wikipedia, today)
- *"Ontology Learning is a mechanism for semi-automatically supporting the ontology engineer in engineering ontologies."*
  A. D. Mädche. Ontology Learning for the Semantic Web. Dissertation. Universität Karlsruhe, 2001
- *"Ontology Learning aims at the integration of a multitude of disciplines in order to facilitate the construction of ontologies, in particular ontology engineering and machine learning."*
  A. D. Mädche, S. Staab. Ontology Learning. Handbook of Ontologies in Information Systems, 2004

Classification of Ontology Learning Data



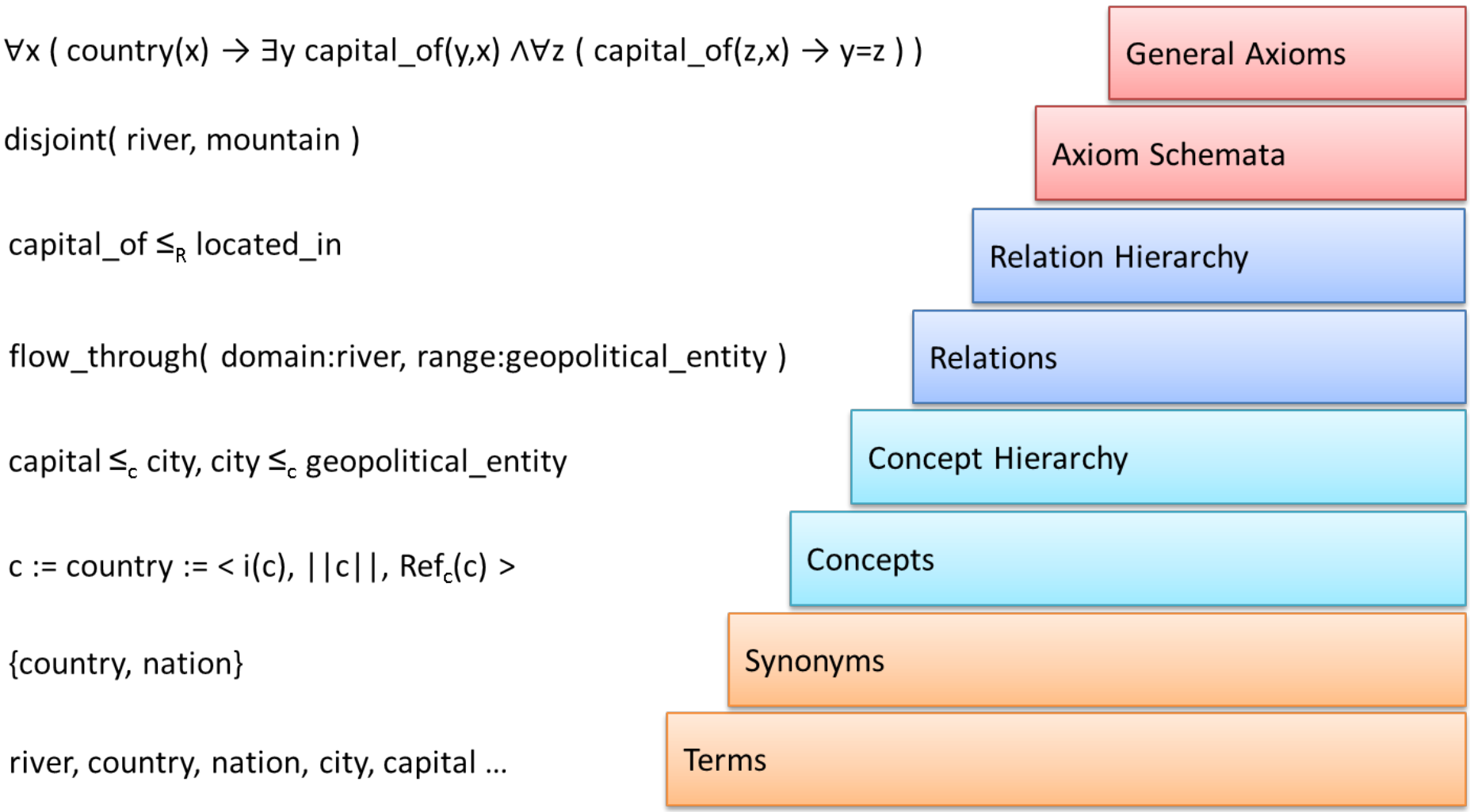
sometimes heterogeneous sources of evidence (e.g., hyponymy [Snow et al. 2006], subsumption [Cimiano et al. 2005], [Manzano-Macho et al. 2008], [Buitelaar et al. 2008], disjointness [Völker et al. 2007])

[Shamsfard, Barforoush 2003]



Outline

Ontology Learning Layer Cake [Cimiano 2006]



Patterns [Hearst 1992] for Class Subsumption

- NP such as {NP,}* {or|and} NP
  - „**games** such as **baseball** and **cricket**“
- NP {,NP}* {,} {and|or} other NP
  - „**rabbits** and other **animals**“
  - but: „**rabbits** and other **pets**“
- NP {,} including {NP,}* {or|and} NP
  - „**fruits** including **apples** and **pears**“
- NP {,} especially {NP,}* {or|and} NP
  - „**Europeans**, especially **Italians**“
  - but: „**US presidents**, especially **democrats**“

Patterns [Ogata and Collier 2004]

- NP is a NP
  - "A **kangaroo** is an **animal** living in Australia."
- a NP named|called NP
  - "Japanese people like to play a **game** called **Go**."
- NP, NP
  - "**Sencha**, the most popular **tea** in Japan, ..."
- NP. The NP
  - "John loves his **Ferrari**. The **car** ..."
- Among NP, NP
  - "Among all **musical instruments**, **violins** are ..."
- NP except for|other than NP
  - "**Employees** except for **managers** suffer from ..."

JAPE Rule

- GATE = General Architecture for Text Engineering
- written in Java
- mature, used worldwide
- JAPE = language for rapid prototyping and efficient implementation of shallow analysis methods
- can be used e.g. for domain specific patterns (financial blogs etc.)

JAPE Rule

```
rule: Hearst_1
(
(NounPhrase):superconcept
{SpaceToken.kind == space}
{Token.string=="such"}
{SpaceToken.kind == space}
{Token.string=="as"}
{SpaceToken.kind == space}
(NounPhrase):subconcept
):hearst1
-->
:hearst1.SubclassOfRelation = { rule = "Hearst1" },
:subconcept.Domain = { rule = "Hearst1" },
:superconcept.Range = { rule = "Hearst1" }
```

Lexical Context Similarity (e.g. [Cimiano and Völker 2005])

- "**Columbus** is the **capital** of the **state** of **Ohio**. **Columbus** has a **population** of about 700,000 **inhabitants**."

Lexical Context Similarity (e.g. [Cimiano and Völker 2005])

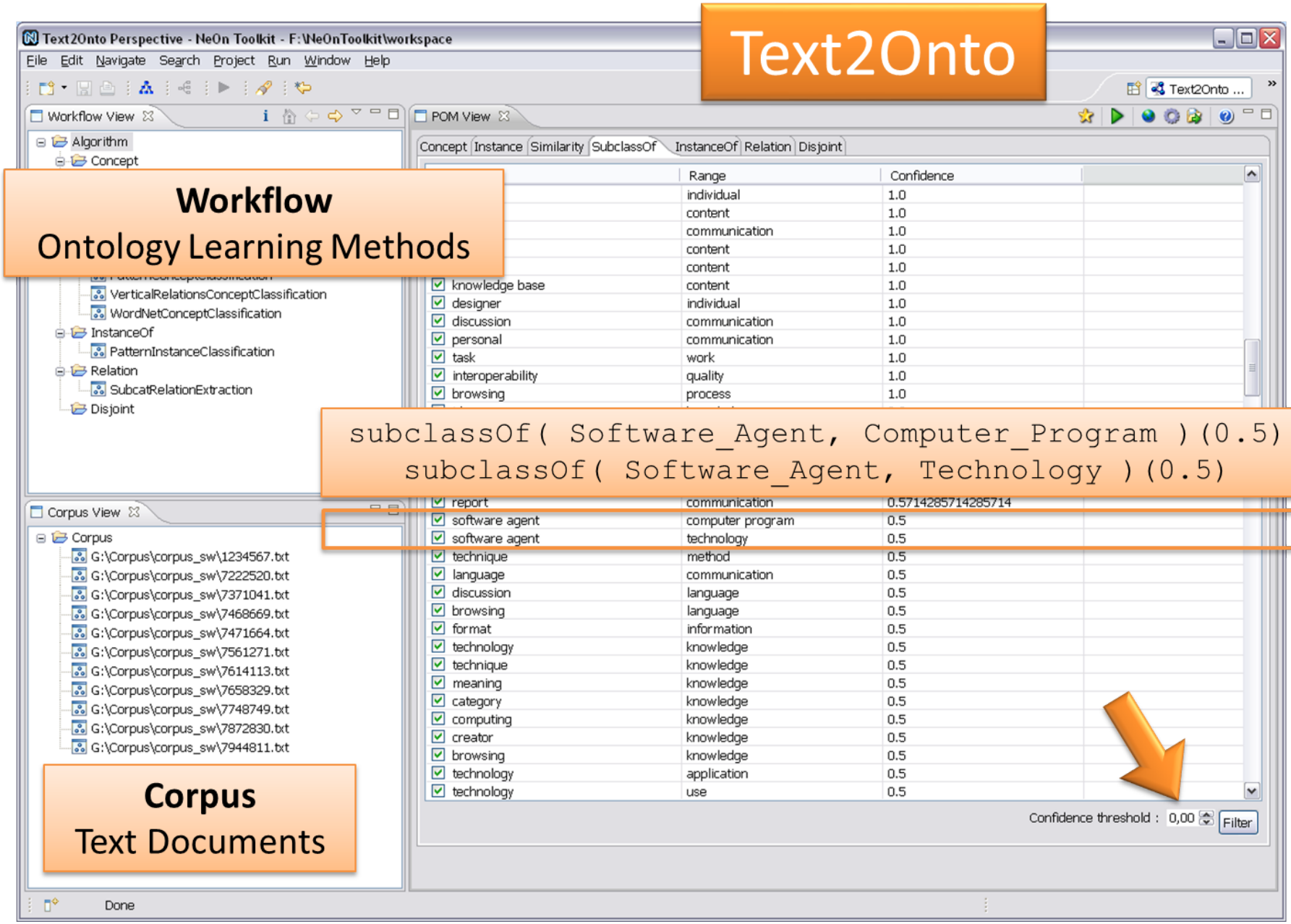
- "**Columbus** is the **capital** of the **state** of **Ohio**. **Columbus** has a **population** of about 700,000 **inhabitants**."
- **Columbus** (capital (1), state (1), Ohio (1), population (1), inhabitant (1))
- City (country (2), state (1), inhabitant (2), mayor (1), attraction (1))
- Explorer (ship (1), sailor (2), discovery (1))

„most probably": City(Columbus)

Subcategorization Frames

- "Tina drives a Ford."
  - Person(Tina). Vehicle(Ford).
- "Her father drives a bus."
  - Father subclass-of Person
  - Bus subclass-of Vehicle
- subcat: drive(subj: person, obj: vehicle)
  - Person $\sqsubseteq \forall$ drive.Vehicle

[Faure and Nédellec 1998], [Schutz and Buitelaar 2005], [Cimiano et al. 2006]


Text2Onto
Text2Onto Perspective - NeOn Toolkit - F:\NeOnToolkit\workspace
File Edit Navigate Search Project Run Window Help
Workflow View
Algorithm
Concept
VerticalRelationsConceptClassification
WordNetConceptClassification
InstanceOf
PatternInstanceClassification
Relation
SubcatRelationExtraction
Disjoint
Workflow
Ontology Learning Methods
POM View
Concept Instance Similarity SubclassOf InstanceOf Relation Disjoint
Range Confidence
individual 1.0
content 1.0
communication 1.0
content 1.0
content 1.0
knowledge base content 1.0
designer individual 1.0
discussion communication 1.0
personal communication 1.0
task work 1.0
interoperability quality 1.0
browsing process 1.0
subclassOf(Software_Agent, Computer_Program)(0.5)
subclassOf(Software_Agent, Technology)(0.5)
report communication 0.5714285714285714
software agent computer program 0.5
software agent technology 0.5
technique method 0.5
language communication 0.5
discussion language 0.5
browsing language 0.5
format information 0.5
technology knowledge 0.5
technique knowledge 0.5
meaning knowledge 0.5
category knowledge 0.5
computing knowledge 0.5
creator knowledge 0.5
browsing knowledge 0.5
technology application 0.5
technology use 0.5
Confidence threshold : 0,00 Filter
Corpus View
Corpus
G:\Corpus\corpus_sw\1234567.txt
G:\Corpus\corpus_sw\7222520.txt
G:\Corpus\corpus_sw\7371041.txt
G:\Corpus\corpus_sw\7468669.txt
G:\Corpus\corpus_sw\7471664.txt
G:\Corpus\corpus_sw\7561271.txt
G:\Corpus\corpus_sw\7614113.txt
G:\Corpus\corpus_sw\7658329.txt
G:\Corpus\corpus_sw\7748749.txt
G:\Corpus\corpus_sw\7872830.txt
G:\Corpus\corpus_sw\7944811.txt
Corpus
Text Documents
Done

Suchanek et al. 2009

Learning from text and background knowledge via reasoning:
"Washington is the capital of the US. (...) New York is the US capital of fashion."

Suchanek et al. 2009

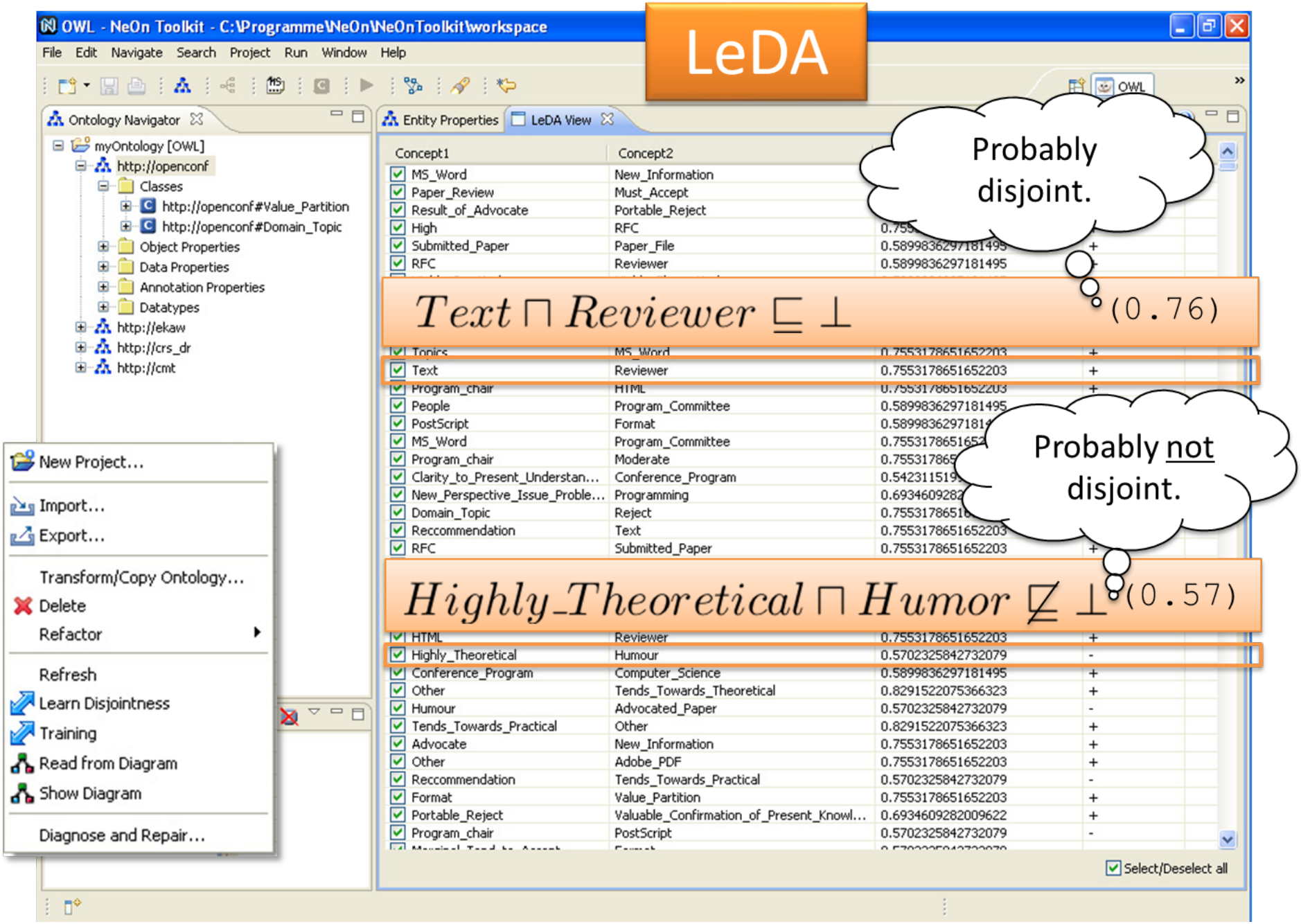
Learning from text and background knowledge via reasoning:
"Washington is the capital of the US. (...) New York is the US capital of fashion."

- extracted: hasCapital(US, New York); hasCapital(US, Washington)
- background knowledge: hasCapital is a functional property

Suchanek et al. 2009

Learning from text and background knowledge via reasoning:
"Washington is the capital of the US. (...) New York is the US capital of fashion."

- extracted: hasCapital(US, New York); hasCapital(US, Washington)
- background knowledge: hasCapital is a functional property
- possible inferences:
  - New York = Washington
  - inconsistency (unique names assumption)
- logical contradictions can help to detect errors in automatically extracted information


LeDA
OWL - NeOn Toolkit - C:\Programme\NeOn\NeOnToolkit\workspace
File Edit Navigate Search Project Run Window Help
Ontology Navigator
myOntology [OWL]
http://openconf
Classes
http://openconf#Value_Partition
http://openconf#Domain_Topic
Object Properties
Data Properties
Annotation Properties
Datatypes
http://ekaw
http://crs_dr
http://cmt
Entity Properties
LeDA View
Concept1
Concept2
MS_Word New_Information
Paper_Review Must_Accept
Result_of_Advocate Portable_Reject
High RFC
Submitted_Paper Paper_File
RFC Reviewer
Probably disjoint.
$Text \sqcap Reviewer \sqsubseteq \bot$ (0.76)
Text Reviewer 0.7553178651652203 +
People Program_Committee 0.5899836297181495 +
PostScript Format
MS_Word Program_Committee
Program_chair Moderate
Clarity_to_Present_Understan... Conference_Program
New_Perspective_Issue_Proble... Programming
Domain_Topic Reject
Reccommendation Text
RFC Submitted_Paper 0.7553178651652203 +
Probably not disjoint.
$Highly\_Theoretical \sqcap Humor \not\sqsubseteq \bot$ (0.57)
HTML Reviewer 0.7553178651652203 +
Highly_Theoretical Humour 0.5702325842732079 -
Conference_Program Computer_Science 0.5899836297181495 +
Other Tends_Towards_Theoretical 0.8291522075366323 +
Humour Advocated_Paper 0.5702325842732079 -
Tends_Towards_Practical Other 0.8291522075366323 +
Advocate New_Information 0.7553178651652203 +
Other Adobe_PDF 0.7553178651652203 +
Reccommendation Tends_Towards_Practical 0.5702325842732079 -
Format Value_Partition 0.7553178651652203 +
Portable_Reject Valuable_Confirmation_of_Present_Knowl... 0.6934609282009622 +
Program_chair PostScript 0.5702325842732079 -
Select/Deselect all
New Project...
Import...
Export...
Transform/Copy Ontology...
Delete
Refactor
Refresh
Learn Disjointness
Training
Read from Diagram
Show Diagram
Diagnose and Repair...

Other Approaches

- Association rules and co-occurrence statistics
- WordNet: hyponymy $\approx$ subsumption
  - hyponym(bank♯1, institution♯1)
  - Bank subclass-of Institution
- Noun phrase heuristics
  - „image processing software“
- Instance clustering (e.g. Columbus and Washington)
  - Hierarchical clustering of context vectors
- Knowledge Base Completion / Formal Concept Analysis (FCA)
  - asks knowledge engineer questions to complete a knowledge base
  - tool: OntoComp [Sertkaya et al.]

Tools and Frameworks

Name	Institute	Authors
ASIUM	INRIA, Jouy-en-Josas	Faure and Nedellec 1999
TextToOnto	AIFB, University of Karlsruhe	Mädche and Volz 2001
HASTI	Amir Kabir University, Teheran	Shamsfard, Barforoush 2004
OntoLT	DFKI, Saarbrücken	Buitelaar et al. 2004
DOODLE	Shizuoka University	Morita et al. 2004
Text2Onto	AIFB, University of Karlsruhe	Cimiano and Völker 2005
OntoLearn	University of Rome	Velardi et al. 2005
OLE	Brno University of Technology	Novacek and Smrz 2005
OntoGen	Institute Jozef Stefan, Ljubljana	Fortuna et al., 2007
GALeOn	Technical University of Madrid	Manzano-Macho et al. 2008
DINO	DERI, Galway	Novacek et al. 2008
OntoLancs	Lancester University	Gacitua et al. 2008

Table: Lexical ontology learning: informal or semi-formal data (e.g. texts)

Tools and Frameworks

Name	Institute	Authors
YINGYANG	University of Bari	Iannone 2006
DL-Learner	University of Leipzig	Lehmann 2006
RELExO	AIFB, University of Karlsruhe	Völker and Rudolph 2008
RoLExO	AIFB, University of Karlsruhe	Völker and Rudolph 2008
OntoComp	University of Dresden	Sertkaya 2008

Table: Logical Ontology Learning

Name	Institute	Authors
LeDA	AIFB, University of Karlsruhe	Völker et al. 2007
SOFIE	MPI, Saarbrücken	Suchanek et al. 2009

Table: Hybrid implementations

Problems and Challenges

- Homonymy and polysemy e.g. [Ovchinnikova et al. 2006]
  - "Peter is sitting on the **bank** in front of the **bank**."
  - "An interesting **book** is lying on the table."
- Semantics of adjectives
  - "**red** flower", "**false** friend"
- Empty heads e.g. [Völker et al. 2005], [Cimiano and Wenderoth 2005]
  - "Tuna is a **kind** of fish. The Southern Bluefin is one of the most endangered **types** of Tuna."
- Ellipsis and underspecification
  - "Mary **started** the book."
- Anaphora (e.g. pronouns) e.g. [Cimiano and Völker 2005]
  - "There is an apple on the table. **It** is red."

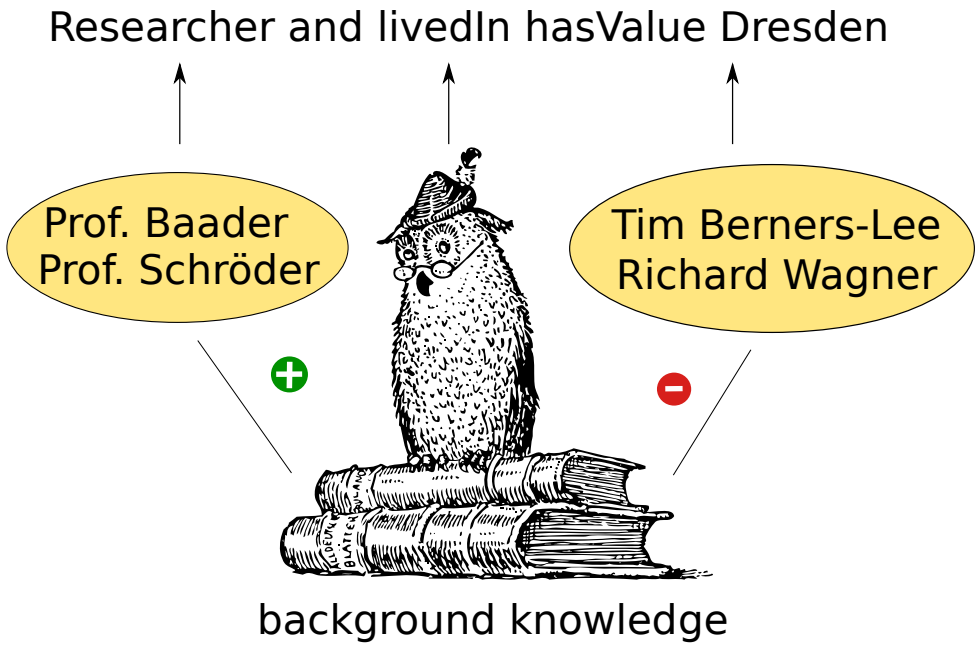
Problems and Challenges (ctd.)

- Metaphors and analogies e.g. [Gust et al. 2007]
  - "**Live** is a **journey**."
- Opinions, quotations and reported speech
  - "Tom thinks that **dolphins** are **mammals**."
- What should be represented as an individual? e.g. [Zirn et al. 2008]
  - "The **kangaroo** is an animal living in Australia."
- Class, relation (object property) or attribute (datatype property)?
  - "All elephants are **grey**."
  - "Easter monday is a **national holiday**."
- Knowledge is changing e.g. [Stojanovic 2004], [Zablith et al. 2009]
  - "**Pluto** is a **planet**."

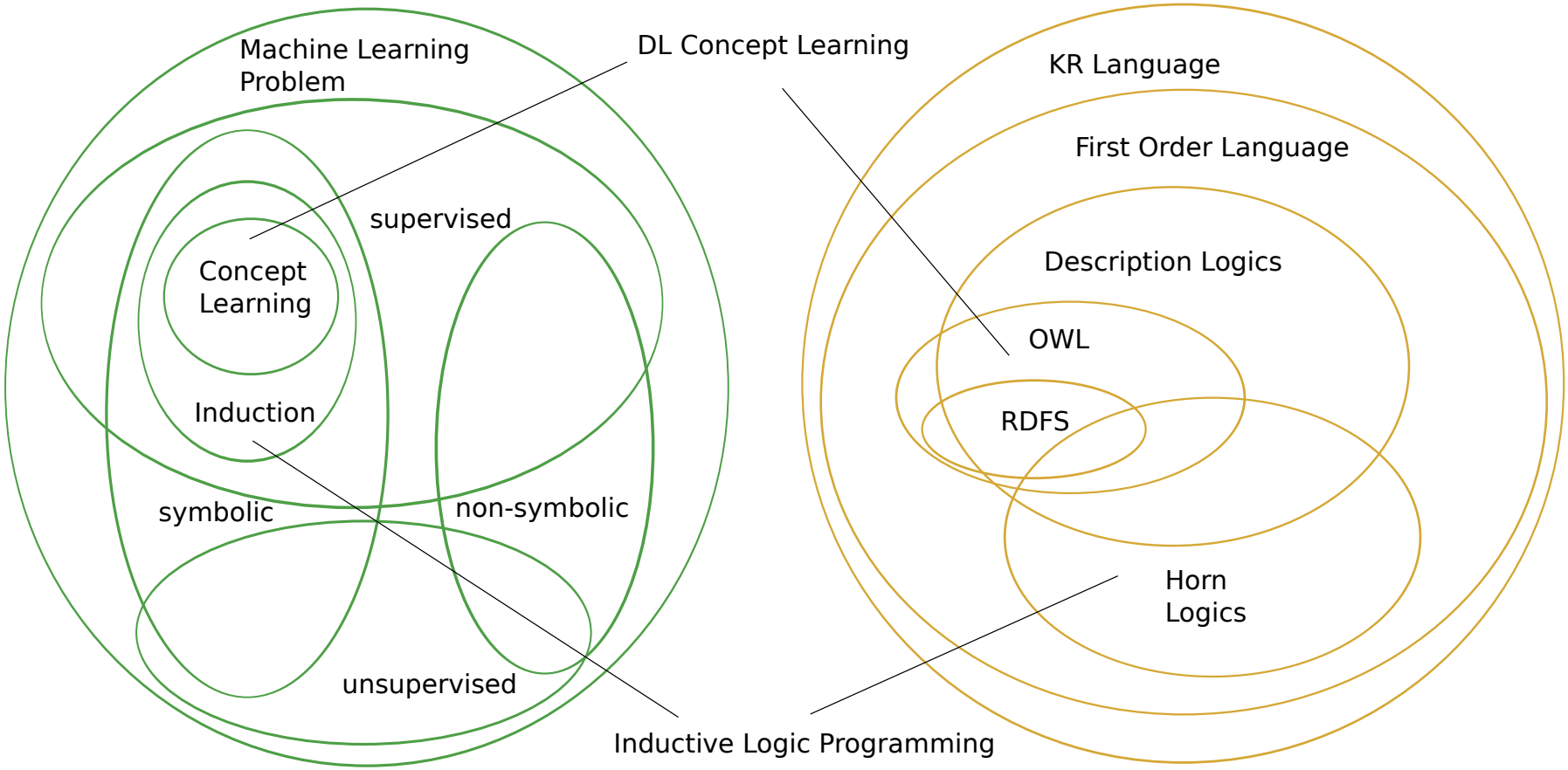
Outline

Learning OWL Class Expressions

- given:
  - background knowledge (particularly OWL/DL knowledge base)
  - positive and negative examples (particulary individuals in knowledge base)
- goal:
  - logical formula (particularly OWL Class Expression) covering positive examples and not covering negative examples



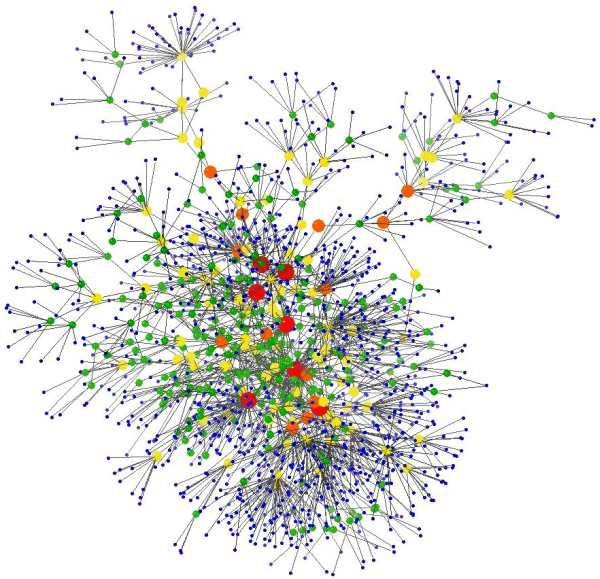
ILP and Semantic Web



- since early 90s Inductive Logic Programming
- only few approaches based on description logics
- Web Ontology Language (OWL) becomes W3C standard in 2004
- increasing number of RDF/OWL knowlegde bases, but ILP still mainly focuses on logic programs $\rightsquigarrow$ research gap

Why ILP in the Semantic Web?

- Ontology Learning:
  - given class A in $\mathcal{K}$
  - instances of A as positive examples
  - non-instances as negative examples
  - definitions can be learned if ABox data is available
- improvement of existing ML problem solutions
- direct usage of knowledge in the Semantic Web instead of conversion in e.g. horn clauses to apply ML methods



ontology network

ML problems

Refinement Operators - Definitions

- given a DL $\mathcal{L}$, consider the quasi-ordered space $\langle \mathcal{C}(\mathcal{L}), \sqsubseteq_{\mathcal{T}} \rangle$ over concepts of $\mathcal{L}$
- $\rho : \mathcal{C}(\mathcal{L}) \to 2^{\mathcal{C}(\mathcal{L})}$ is a downward $\mathcal{L}$ refinement operator if for any $C \in \mathcal{C}(\mathcal{L})$:

$$D \in \rho(C) \text{ implies } D \sqsubseteq_{\mathcal{T}} C$$

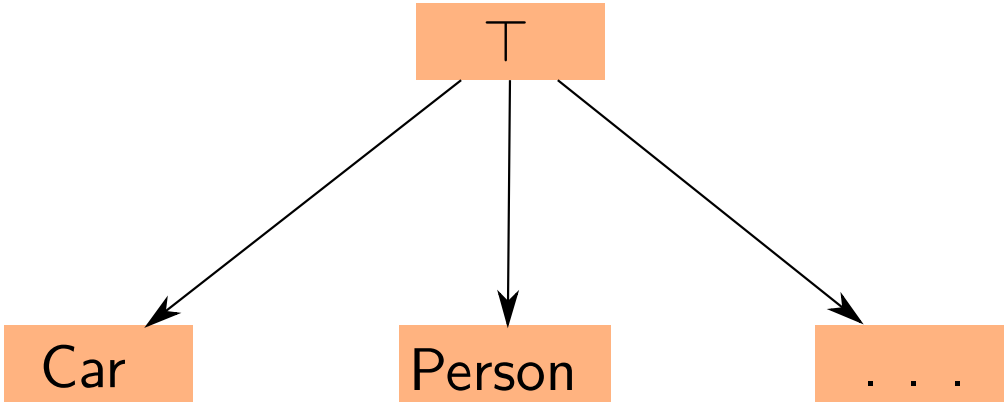
- notation: Write $C \rightsquigarrow_{\rho} D$ instead of $D \in \rho(C)$
- example refinement chain in $\langle \mathcal{C}(\mathcal{EL}), \sqsubseteq_{\mathcal{T}} \rangle$:

$$\top \rightsquigarrow_{\rho} \texttt{Male} \rightsquigarrow_{\rho} \texttt{Male} \sqcap \exists \texttt{hasChild}.\top$$

Learning with Refinement Operators

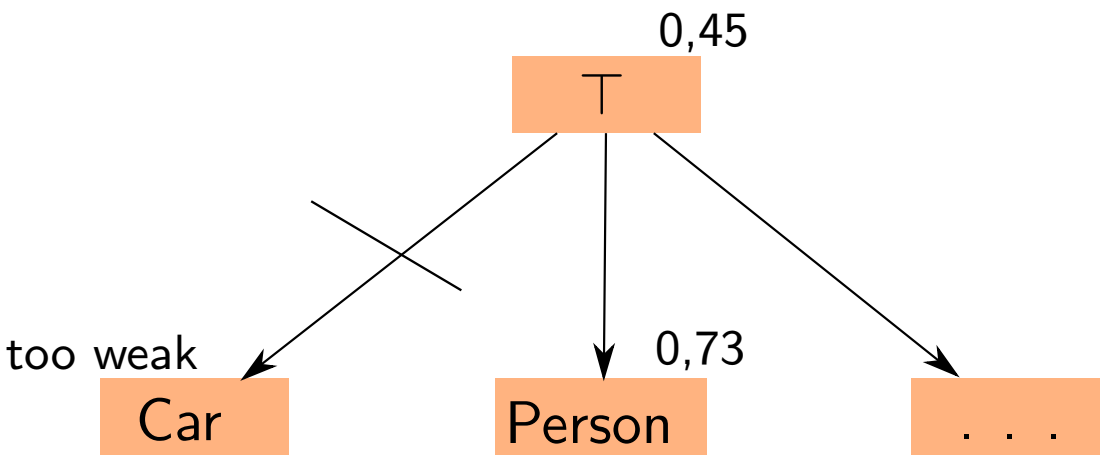
- start with most general concept (top)

Learning with Refinement Operators



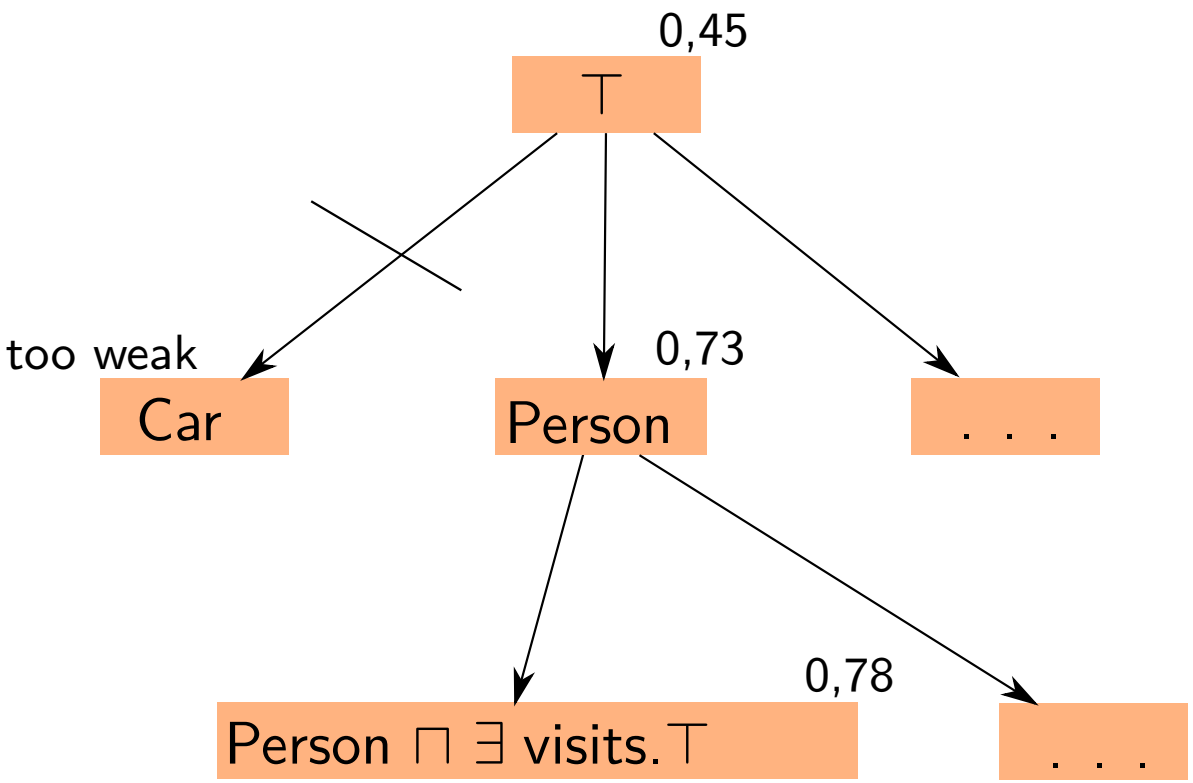
- start with most general concept (top)
- operator specialises concept

Learning with Refinement Operators



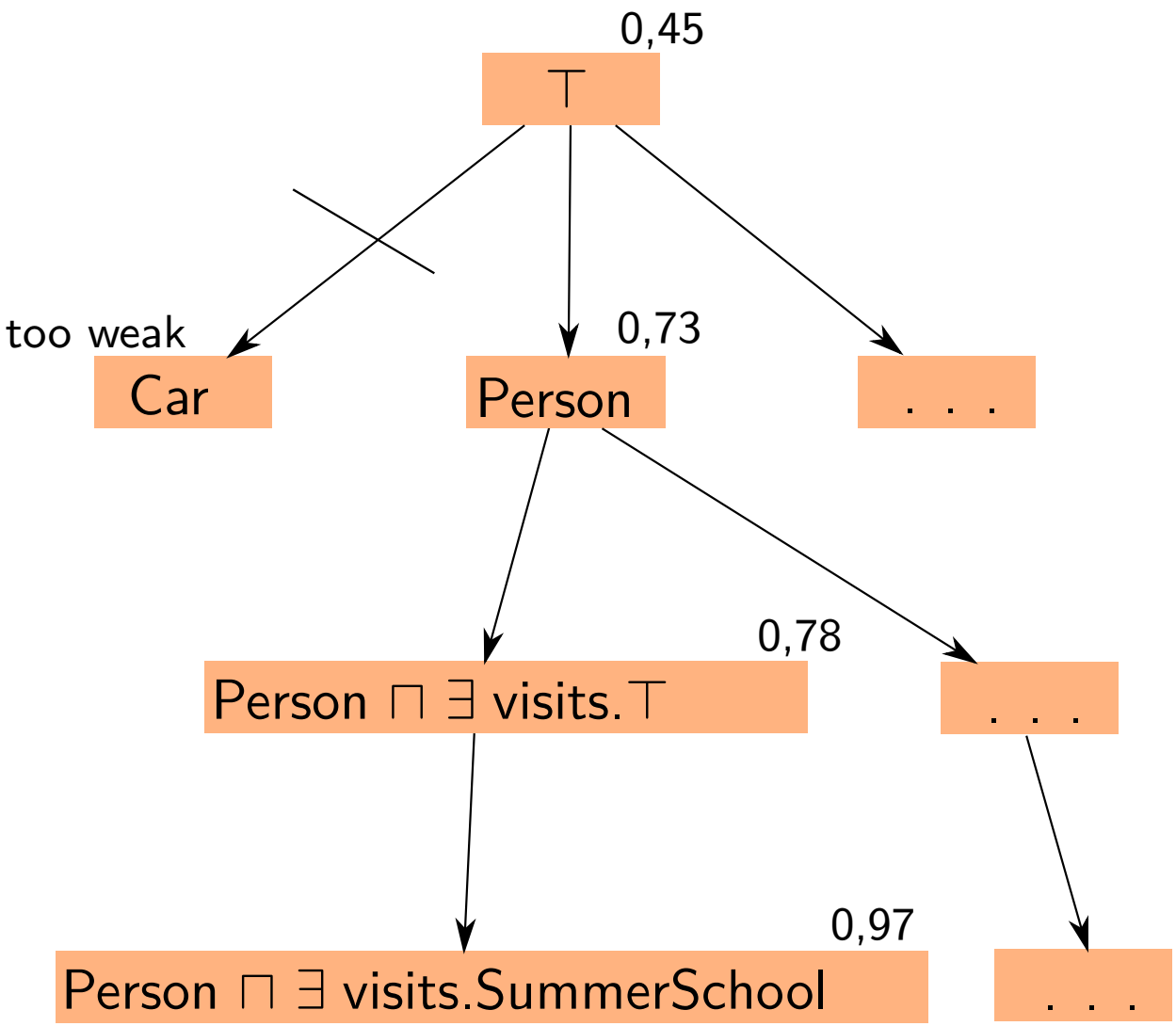
- start with most general concept (top)
- operator specialises concept
- heuristic assigns score using pos/neg examples

Learning with Refinement Operators



- start with most general concept (top)
- operator specialises concept
- heuristic assigns score using pos/neg examples
- continue until termination criterion is met

Learning with Refinement Operators



- start with most general concept (top)
- operator specialises concept
- heuristic assigns score using pos/neg examples
- continue until termination criterion is met

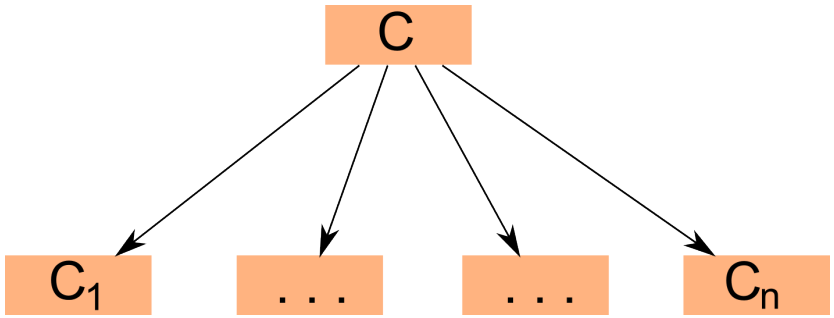
=

learning algorithm

Properties of Refinement Operators

An $\mathcal{L}$ downward refinement operator ρ is called

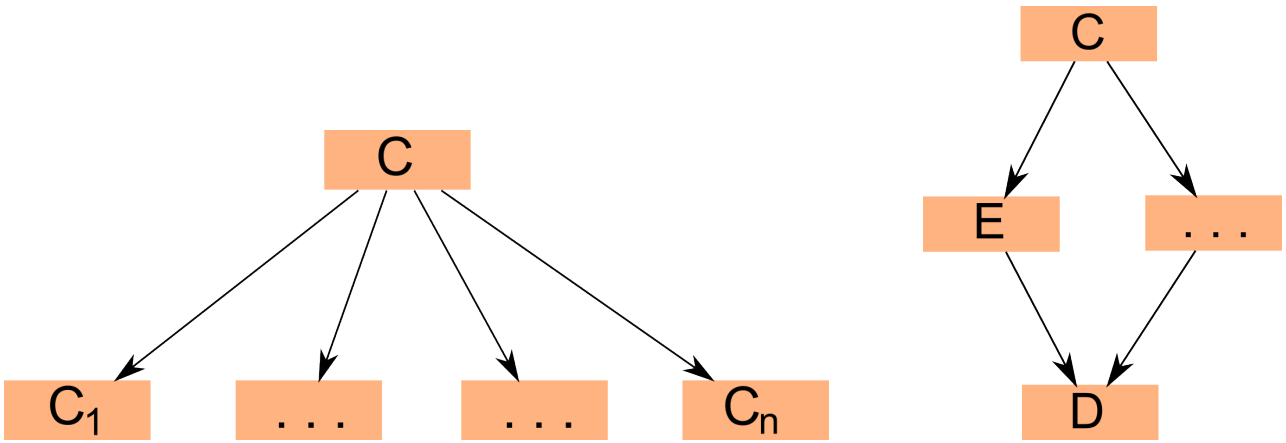
- finite iff $\rho(C)$ is finite for any concept $C \in \mathcal{C}(\mathcal{L})$



Properties of Refinement Operators

An $\mathcal{L}$ downward refinement operator ρ is called

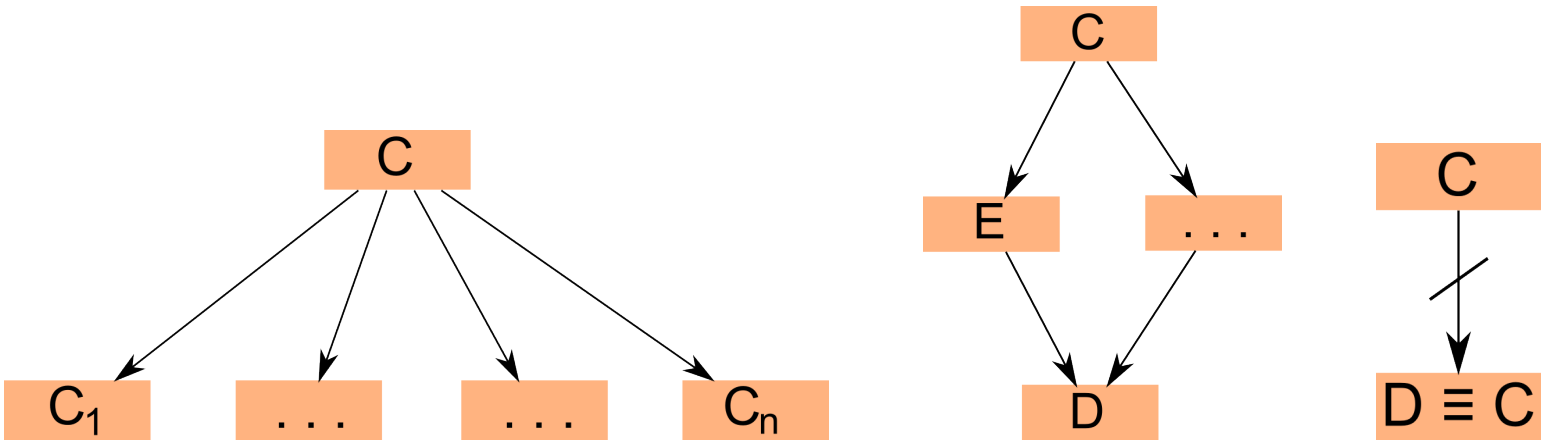
- finite iff $\rho(C)$ is finite for any concept $C \in \mathcal{C}(\mathcal{L})$
- redundant iff there exist two different ρ refinement chains from a concept C to a concept D.



Properties of Refinement Operators

An $\mathcal{L}$ downward refinement operator ρ is called

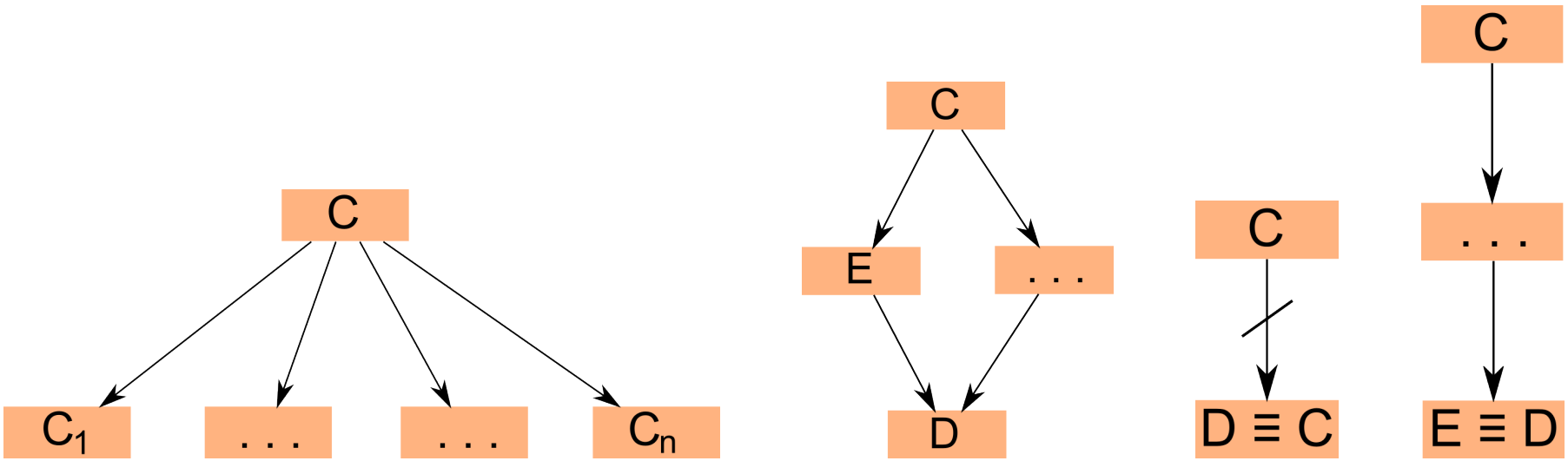
- finite iff $\rho(C)$ is finite for any concept $C \in \mathcal{C}(\mathcal{L})$
- redundant iff there exist two different ρ refinement chains from a concept C to a concept D.
- proper iff for $C, D \in \mathcal{C}(\mathcal{L})$, $C \rightsquigarrow_\rho D$ implies $C \not\equiv_\mathcal{T} D$



Properties of Refinement Operators

An $\mathcal{L}$ downward refinement operator ρ is called

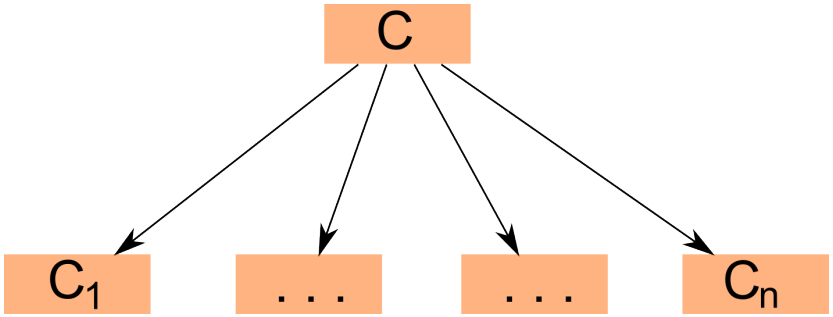
- finite iff $\rho(C)$ is finite for any concept $C \in \mathcal{C}(\mathcal{L})$
- redundant iff there exist two different ρ refinement chains from a concept C to a concept D.
- proper iff for $C, D \in \mathcal{C}(\mathcal{L})$, $C \rightsquigarrow_\rho D$ implies $C \not\equiv_\mathcal{T} D$
- complete iff for $C, D \in \mathcal{C}(\mathcal{L})$ with $D \sqsubset_\mathcal{T} C$ there is a concept E with $E \equiv_\mathcal{T} D$ and a refinement chain $C \rightsquigarrow_\rho \cdots \rightsquigarrow_\rho E$
- weakly complete iff for any concept C with $C \sqsubset_\mathcal{T} \top$ we can reach a concept E with $E \equiv_\mathcal{T} C$ from $\top$ by ρ.

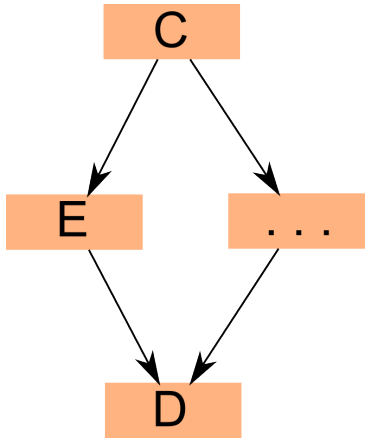


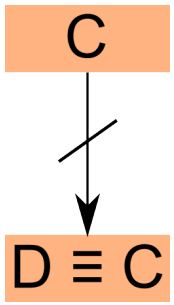
Properties of Refinement Operators

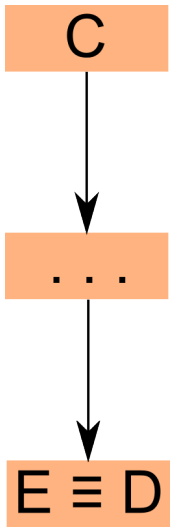
An $\mathcal{L}$ downward refinement operator ρ is called

- finite iff $\rho(C)$ is finite for any concept $C \in \mathcal{C}(\mathcal{L})$
- redundant iff there exist two different ρ refinement chains from a concept C to a concept D.
- proper iff for $C, D \in \mathcal{C}(\mathcal{L})$, $C \rightsquigarrow_\rho D$ implies $C \not\equiv_\mathcal{T} D$
- complete iff for $C, D \in \mathcal{C}(\mathcal{L})$ with $D \sqsubset_\mathcal{T} C$ there is a concept E with $E \equiv_\mathcal{T} D$ and a refinement chain $C \rightsquigarrow_\rho \cdots \rightsquigarrow_\rho E$
- weakly complete iff for any concept C with $C \sqsubset_\mathcal{T} \top$ we can reach a concept E with $E \equiv_\mathcal{T} C$ from $\top$ by ρ.
- ideal = complete + proper + finite









Properties of Refinement Operators

- Properties indicate how suitable a refinement operator is for solving the learning problem:
  - Incomplete operators may miss solutions
  - Redundant operators may lead to duplicate concepts in the search tree
  - Improper operators may produce equivalent concepts (which cover the same examples)
  - For infinite operators it may not be possible to compute all refinements of a given concept
- We researched properties of refinement operators in Description Logics
- Key question: Which properties can be combined?

Refinement Operator Property Theorem

Theorem

Maximal sets of properties of $\mathcal{L}$ refinement operators which can be combined for $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCN}, \mathcal{SHOIN}, \mathcal{SROIQ}\}$:

1. {*weakly complete*, *complete*, *finite*}
2. {*weakly complete*, *complete*, *proper*}
3. {*weakly complete*, *non-redundant*, *finite*}
4. {*weakly complete*, *non-redundant*, *proper*}
5. {*non-redundant*, *finite*, *proper*}

"Foundations of Refinement Operators for Description Logics", *J. Lehmann, P. Hitzler, ILP conference, 2008*

"Concept Learning in Description Logics Using Refinement Operators", *J. Lehmann, P. Hitzler, Machine Learning journal, 2010*

Refinement Operator Property Theorem

- no ideal refinement in OWL and many description logics
- indicates that learning in DLs is hard
- algorithms need to counteract disadvantages
- goal: develop operators close to theoretical limits

Definition of ρ

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

$$\rho_B(C) = \begin{cases}
\emptyset & \text{if } C = \bot \\
\{C_1 \sqcup \cdots \sqcup C_n \mid C_i \in M_B\ (1 \leq i \leq n)\} & \text{if } C = \top \\
\{A' \mid A' \in sh_\downarrow(A)\} & \text{if } C = A\ (A \in N_C) \\
\quad \cup \{A \sqcap D \mid D \in \rho_B(\top)\} & \\
\{\neg A' \mid A' \in sh_\uparrow(A)\} & \text{if } C = \neg A\ (A \in N_C) \\
\quad \cup \{\neg A \sqcap D \mid D \in \rho_B(\top)\} & \\
\{\exists r.E \mid A = ar(r), E \in \rho_A(D)\} & \text{if } C = \exists r.D \\
\quad \cup \{\exists r.D \sqcap E \mid E \in \rho_B(\top)\} & \\
\quad \cup \{\exists s.D \mid s \in sh_\downarrow(r)\} & \\
\{\forall r.E \mid A = ar(r), E \in \rho_A(D)\} & \text{if } C = \forall r.D \\
\quad \cup \{\forall r.D \sqcap E \mid E \in \rho_B(\top)\} & \\
\quad \cup \{\forall r.\bot \mid & \\
\qquad D = A \in N_C \text{ and } sh_\downarrow(A) = \emptyset\} & \\
\quad \cup \{\forall s.D \mid s \in sh_\downarrow(r)\} & \\
\{C_1 \sqcap \cdots \sqcap C_{i-1} \sqcap D \sqcap C_{i+1} \sqcap \cdots \sqcap C_n \mid & \text{if } C = C_1 \sqcap \cdots \sqcap C_n \\
\qquad D \in \rho_B(C_i), 1 \leq i \leq n\} & (n \geq 2) \\
\{C_1 \sqcup \cdots \sqcup C_{i-1} \sqcup D \sqcup C_{i+1} \sqcup \cdots \sqcup C_n \mid & \text{if } C = C_1 \sqcup \cdots \sqcup C_n \\
\qquad D \in \rho_B(C_i), 1 \leq i \leq n\} & (n \geq 2) \\
\quad \cup \{(C_1 \sqcup \cdots \sqcup C_n) \sqcap D \mid & \\
\qquad D \in \rho_B(\top)\} &
\end{cases}$$

Base Operator (excerpt)

Definition of ρ

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

$$\rho_B(C) = \begin{cases}
\emptyset & \text{if } C = \bot \\
\{C_1 \sqcup \cdots \sqcup C_n \mid C_i \in M_B \ (1 \leq i \leq n)\} & \text{if } C = \top \\
\{A' \mid A' \in sh_\downarrow(A)\} & \text{if } C = A \ (A \in N_C) \\
\quad \cup \{A \sqcap D \mid D \in \rho_B(\top)\} & \\
\{\neg A' \mid A' \in sh_\uparrow(A)\} & \text{if } C = \neg A \ (A \in N_C) \\
\quad \cup \{\neg A \sqcap D \mid D \in \rho_B(\top)\} & \\
\{\exists r.E \mid A = ar(r), E \in \rho_A(D)\} & \text{if } C = \exists r.D \\
\quad \cup \{\exists r.D \sqcap E \mid E \in \rho_B(\top)\} & \\
\quad \cup \{\exists s.D \mid s \in sh_\downarrow(r)\} & \\
\{\forall r.E \mid A = ar(r), E \in \rho_A(D)\} & \text{if } C = \forall r.D \\
\quad \cup \{\forall r.D \sqcap E \mid E \in \rho_B(\top)\} & \\
\quad \cup \{\forall r.\bot \mid & \\
\qquad D = A \in N_C \text{ and } sh_\downarrow(A) = \emptyset\} & \\
\quad \cup \{\forall s.D \mid s \in sh_\downarrow(r)\} & \\
\{C_1 \sqcap \cdots \sqcap C_{i-1} \sqcap D \sqcap C_{i+1} \sqcap \cdots \sqcap C_n \mid & \text{if } C = C_1 \sqcap \cdots \sqcap C_n \\
\qquad D \in \rho_B(C_i), 1 \leq i \leq n\} & (n \geq 2) \\
\{C_1 \sqcup \cdots \sqcup C_{i-1} \sqcup D \sqcup C_{i+1} \sqcup \cdots \sqcup C_n \mid & \text{if } C = C_1 \sqcup \cdots \sqcup C_n \\
\qquad D \in \rho_B(C_i), 1 \leq i \leq n\} & (n \geq 2) \\
\quad \cup \{(C_1 \sqcup \cdots \sqcup C_n) \sqcap D \mid & \\
\qquad D \in \rho_B(\top)\} &
\end{cases}$$

Base Operator (excerpt)

Definition of ρ

$$\{\exists r.E \mid A = ar(r), E \in \rho_A(D)\} \quad \text{if } C = \exists r.D$$
$$\cup\ \{\exists r.D \sqcap E \mid E \in \rho_B(\top)\}$$
$$\cup\ \{\exists s.D \mid s \in sh_{\downarrow}(r)\}$$

Examples:

$$\exists \texttt{takesPartIn.SocialGathering} \rightsquigarrow \exists \texttt{takesPartIn.Meeting}$$

Definition of ρ

$$\{\exists r.E \mid A = ar(r), E \in \rho_A(D)\} \qquad \text{if } C = \exists r.D$$
$$\cup\ \{\exists r.D \sqcap E \mid E \in \rho_B(\top)\}$$
$$\cup\ \{\exists s.D \mid s \in sh_{\downarrow}(r)\}$$

Examples:

$$\exists \texttt{takesPartIn.SocialGathering} \rightsquigarrow \exists \texttt{takesPartIn.Meeting}$$
$$\texttt{Student} \sqcap \exists \texttt{takesPartIn.SocialGathering}$$

Definition of ρ

$$\{\exists r.E \mid A = ar(r), E \in \rho_A(D)\} \qquad \text{if } C = \exists r.D$$
$$\cup\ \{\exists r.D \sqcap E \mid E \in \rho_B(\top)\}$$
$$\cup\ \{\exists s.D \mid s \in sh_{\downarrow}(r)\}$$

Examples:

$$\texttt{∃takesPartIn.SocialGathering} \rightsquigarrow$$
$$\texttt{∃takesPartIn.Meeting}$$
$$\texttt{Student} \sqcap \texttt{∃takesPartIn.SocialGathering}$$
$$\texttt{∃leads.SocialGathering}$$

ρ Properties

- $\rho_\downarrow$ is complete
- $\rho_\downarrow$ is infinite, e.g. there are infinitely many refinement steps of the form:

$$\top \rightsquigarrow_{\rho_\downarrow} C_1 \sqcup C_2 \sqcup C_3 \sqcup \ldots$$

- $\rho_\downarrow$ not proper, but can be extended to a proper operator $\rho_\downarrow^{cl}$ (refinements more expensive to compute)
- $\rho_\downarrow$ is redundant:

$$\begin{array}{ccc} \forall r_1.A_1 \sqcup \forall r_2.A_1 & \rightsquigarrow_{\rho_\downarrow} & \forall r_1.(A_1 \sqcap A_2) \sqcup \forall r_2.A_1 \\ \upharpoonleft_{\rho_\downarrow} & & \upharpoonleft_{\rho_\downarrow} \\ \forall r_1.A_1 \sqcup \forall r_2.(A_1 \sqcap A_2) & \rightsquigarrow_{\rho_\downarrow} & \forall r_1.(A_1 \sqcap A_2) \sqcup \forall r_2.(A_1 \sqcap A_2) \end{array}$$

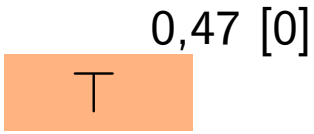
"A Refinement Operator Based Learning Algorithm for the $\mathcal{ALC}$ Description Logic", J. Lehmann, P. Hitzler, ILP conference, 2008

"Concept Learning in Description Logics Using Refinement Operators", J. Lehmann, P. Hitzler, Machine Learning journal, 2010

OCEL

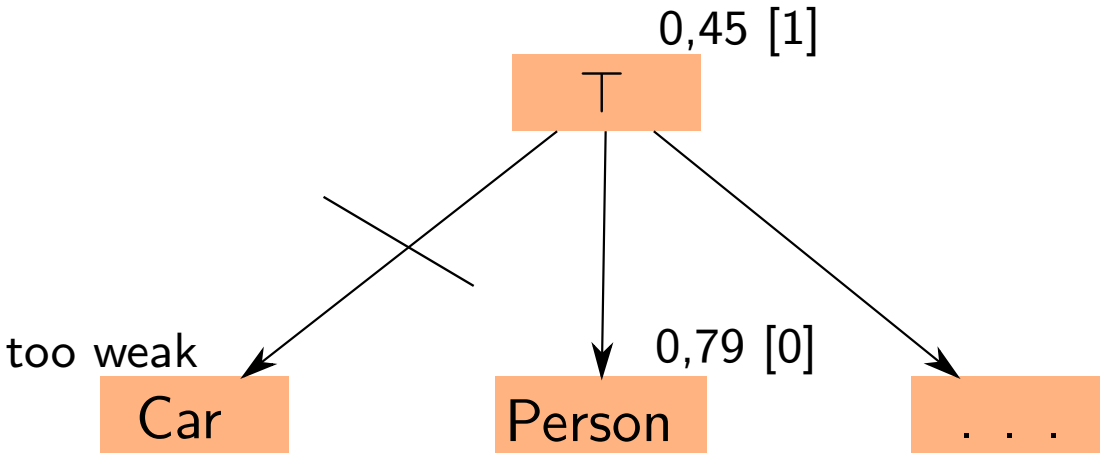
- uses ρ for top down search
- OCEL is complete - it always find a solution if one exists
- highly configurable, e.g. flexible target language, termination criteria and heuristics
- implements redundancy elimination technique with polynomial complexity wrt. search tree size based on ordered negation normal form
- can handle infinite refinement operators by stepwise length-limited horizontal expansion

Stepwise Node Expansion



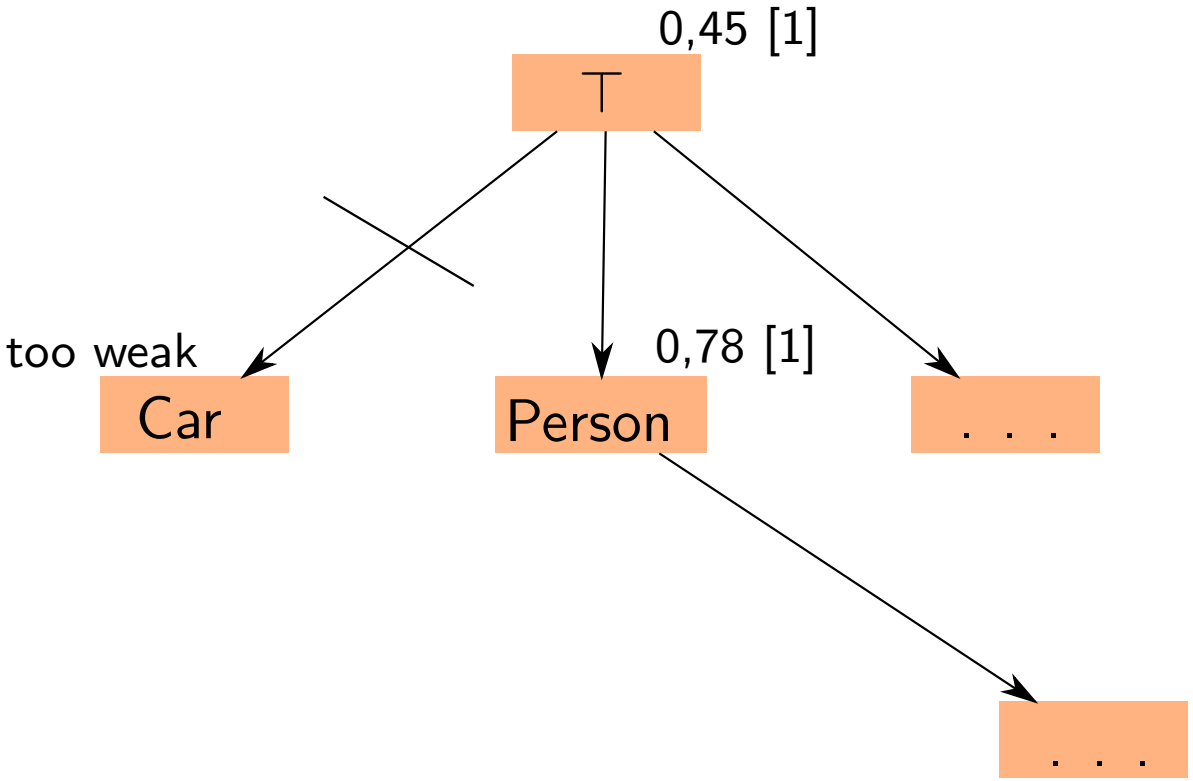
- length of child concepts limited by horesuchtizontal expansion (he)

Stepwise Node Expansion



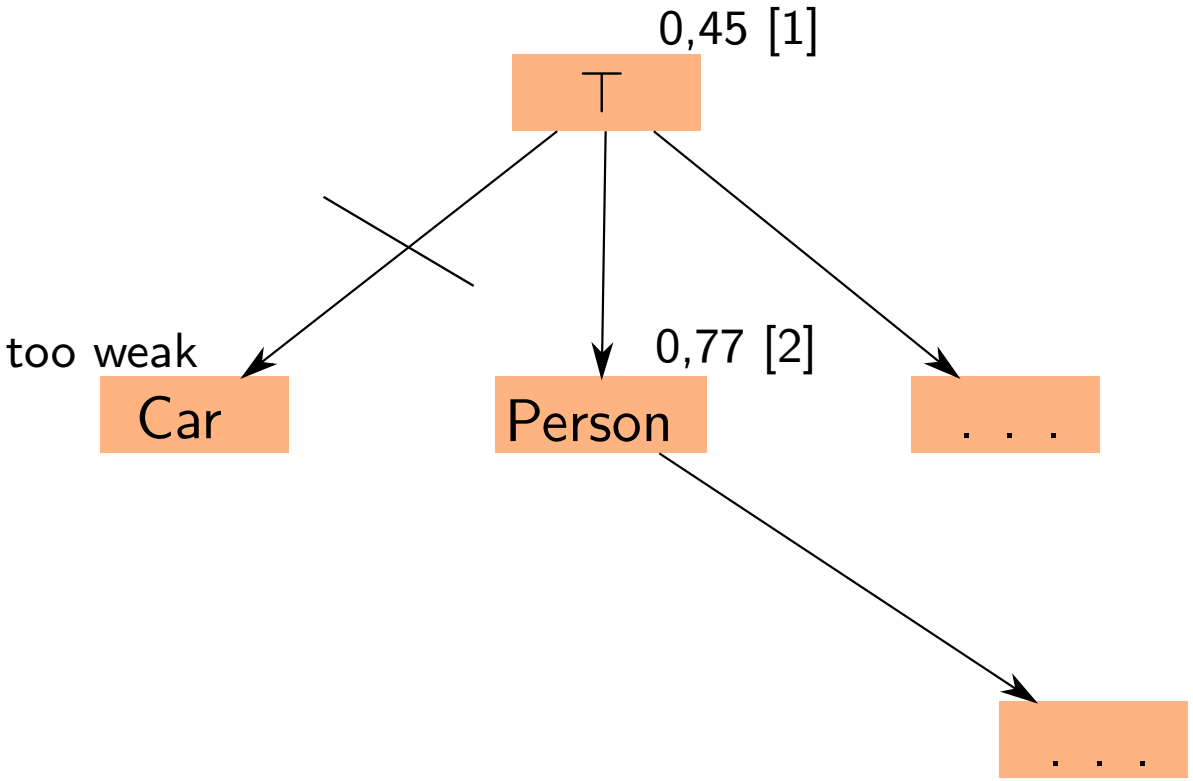
- length of child concepts limited by horizontal expansion (he)
- ρ (infinite) is applicable

Stepwise Node Expansion



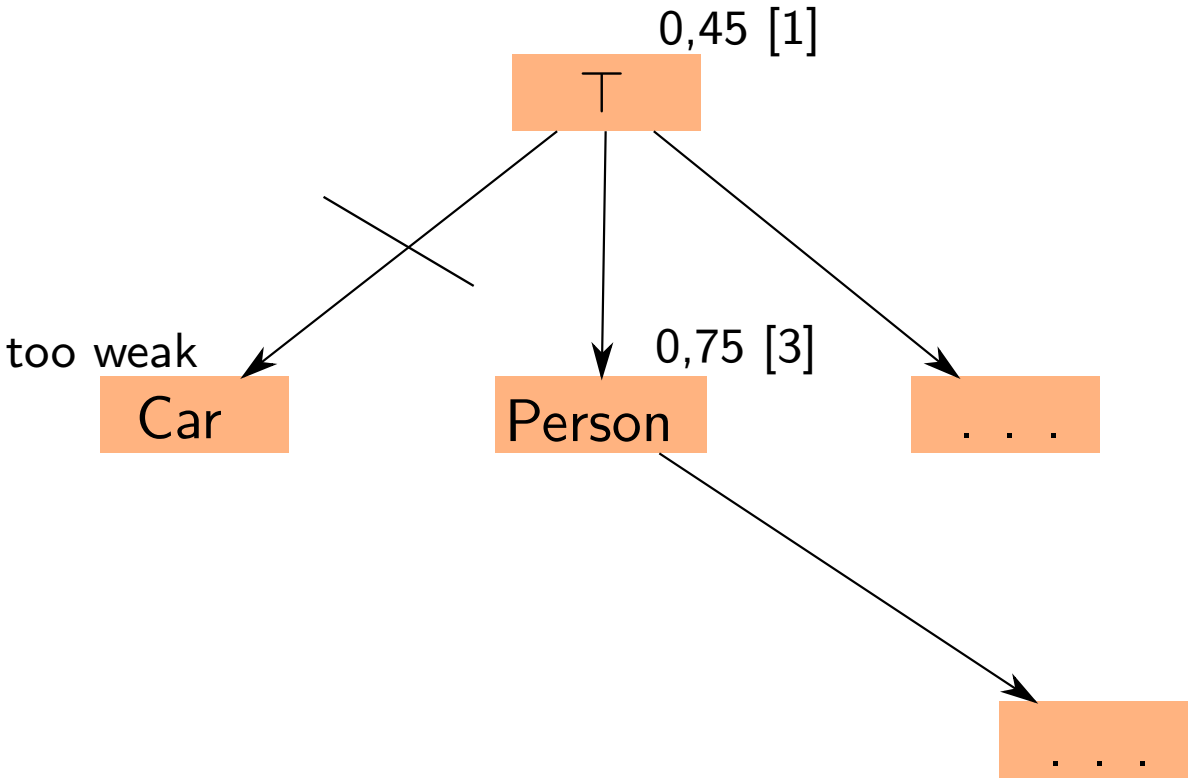
- length of child concepts limited by horizontal expansion (he)
- ρ (infinite) is applicable
- he influences heuristic (Bias towards short concepts - Occam's Razor, higher diversity)

Stepwise Node Expansion



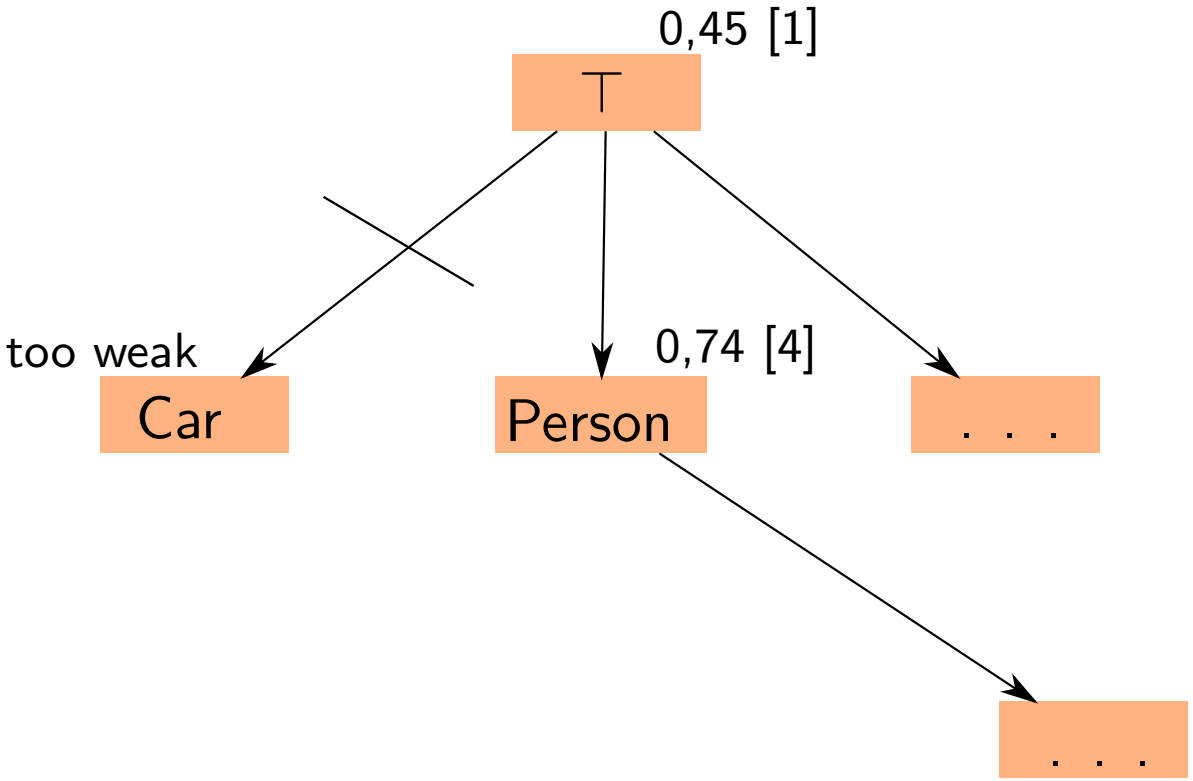
- length of child concepts limited by horizontal expansion (he)
- ρ (infinite) is applicable
- he influences heuristic (Bias towards short concepts - Occam's Razor, higher diversity)

Stepwise Node Expansion



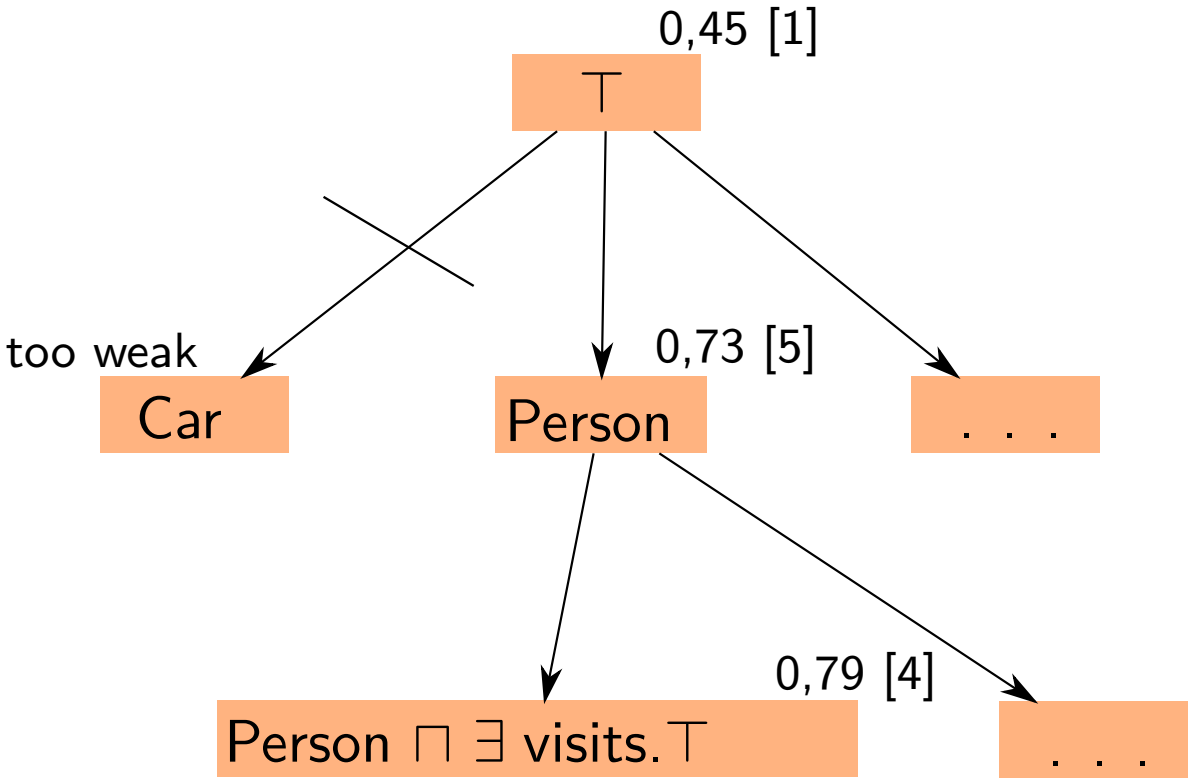
- length of child concepts limited by horizontal expansion (he)
- ρ (infinite) is applicable
- he influences heuristic (Bias towards short concepts - Occam's Razor, higher diversity)

Stepwise Node Expansion



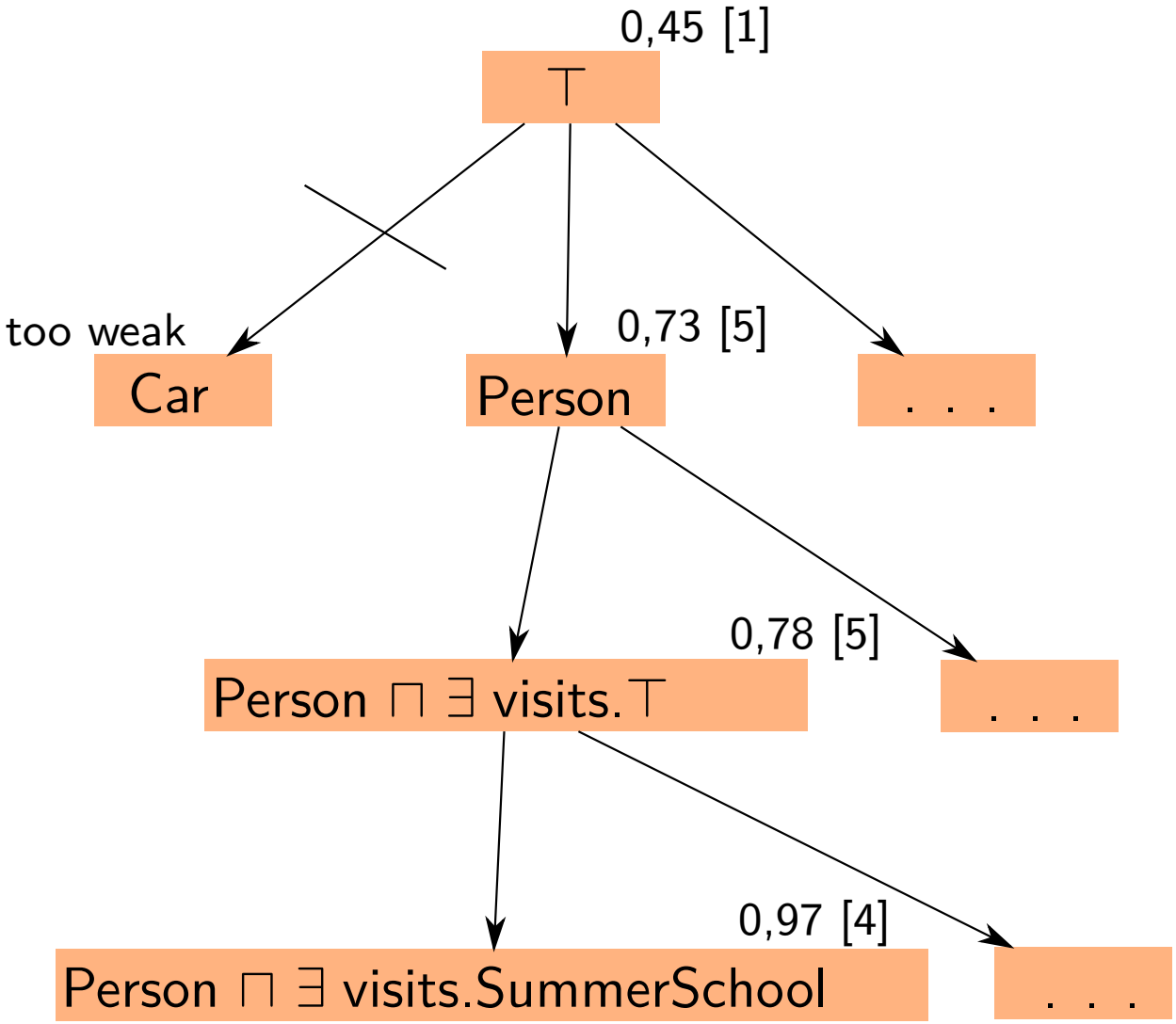
- length of child concepts limited by horizontal expansion (he)
- ρ (infinite) is applicable
- he influences heuristic (Bias towards short concepts - Occam's Razor, higher diversity)

Stepwise Node Expansion



- length of child concepts limited by horizontal expansion (he)
- ρ (infinite) is applicable
- he influences heuristic (Bias towards short concepts - Occam's Razor, higher diversity)

Stepwise Node Expansion



- length of child concepts limited by horizontal expansion (he)
- ρ (infinite) is applicable
- he influences heuristic (Bias towards short concepts - Occam's Razor, higher diversity)

Scalability: Reasoning

$$\mathcal{K} = \{\texttt{Male} \sqsubseteq \texttt{Person},$$
$$\texttt{OnlyMaleChildren}(a),$$
$$\texttt{Person}(a), \texttt{Male}(a_1), \texttt{Male}(a_2),$$
$$\texttt{hasChild}(a, a_1), \texttt{hasChild}(a, a_2)\}$$

- given $\mathcal{K}$, we want to learn a description of `OnlyMaleChildren`
- $C = \texttt{Person} \sqcap \forall \texttt{hasChild}.\texttt{Male}$ appears to be a good solution, but a is not an instance of C under OWA
- idea: dematerialise $\mathcal{K}$ using standard (OWA) DL reasoner, but perform instance checks using CWA
- closer to intuition and provides order of magnitude performance improvements
- optimised for thousands of instance checks on a static knowledge base

Scalability: Stochastic Coverage Computation

Heuristics often require expensive instance checks or retrieval, e.g.:

$$\frac{1}{2} \cdot \left(\frac{|R(A) \cap R(C)|}{|R(A)|} + \sqrt{\frac{|R(A) \cap R(C)|}{|R(C)|}} \right)$$

Scalability: Stochastic Coverage Computation

Heuristics often require expensive instance checks or retrieval, e.g.:

$$\frac{1}{2} \cdot \left(\frac{a}{|R(A)|} + \sqrt{\frac{a}{b}} \right)$$

- replace $|R(A) \cap R(C)|$ und $|R(C)|$ by variables a and b we want to estimate
- Wald-Method for computing the 95% confidence interval
- first estimate a, then the whole expressions
- method can be applied to various heuristics

Scalability: Stochastic Coverage Computation

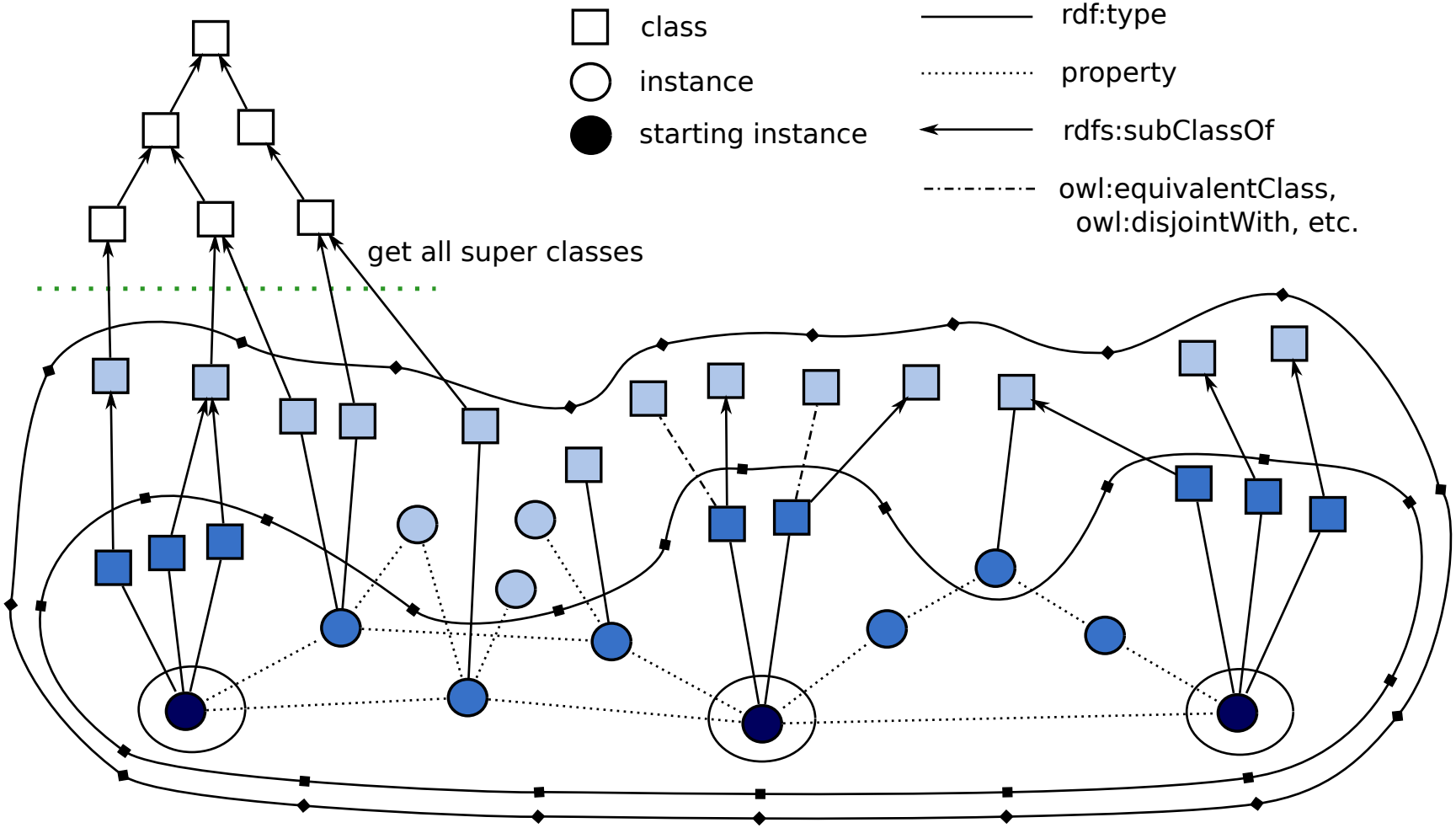
Heuristics often require expensive instance checks or retrieval, e.g.:

$$\frac{1}{2} \cdot \left(\frac{a}{|R(A)|} + \sqrt{\frac{a}{b}} \right)$$

- replace $|R(A) \cap R(C)|$ und $|R(C)|$ by variables a and b we want to estimate
- Wald-Method for computing the 95% confidence interval
- first estimate a, then the whole expressions
- method can be applied to various heuristics
- in tests on real ontologies up to 99% less instance checks and algorithm up to 30 times faster
- low influence on learning results empirically shown in 380 learning problems on 7 real ontologies (differs by ca. $0,2\% \pm 0,4\%$)

Scalability: Fragment Extraction

Extraction of Fragments from SPARQL Endpoints / Linked Data:

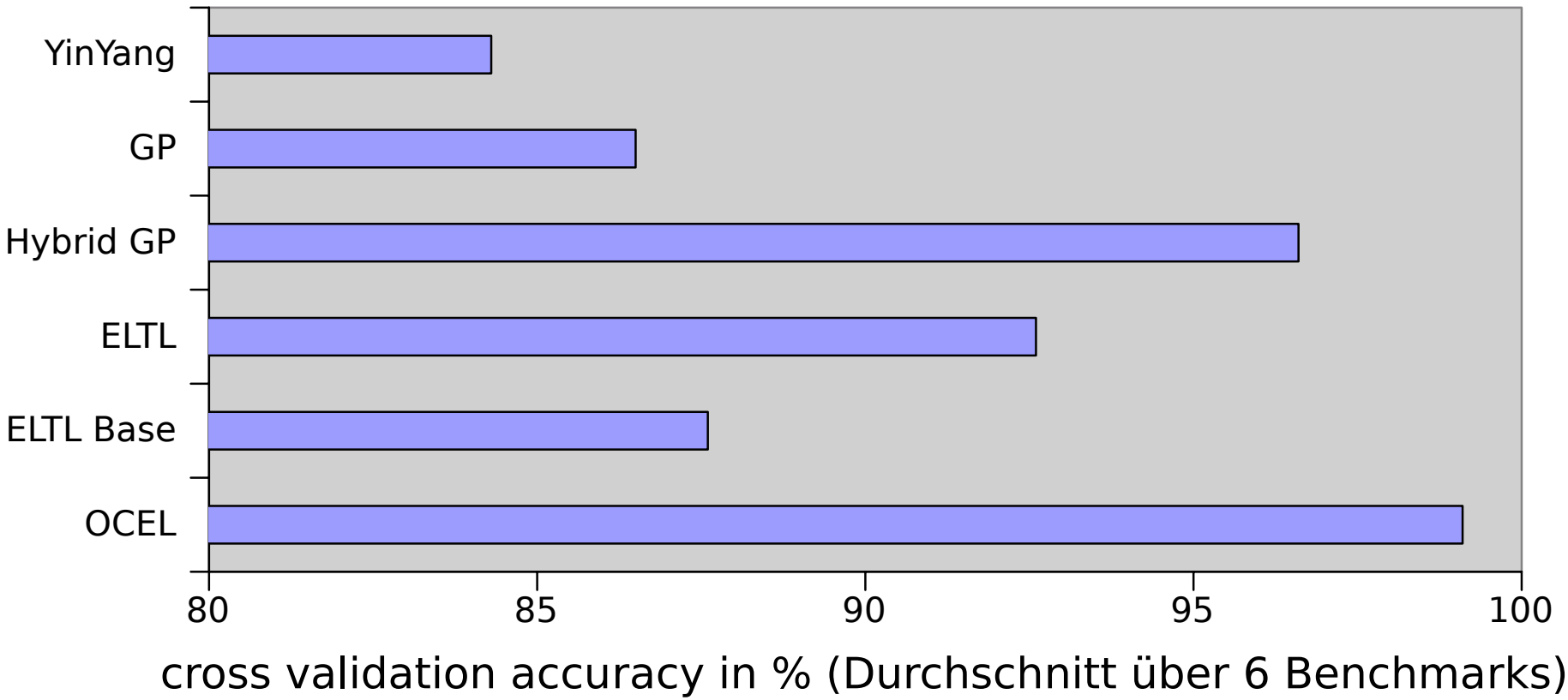


"Learning of OWL Class Descriptions on Very Large Knowledge Bases", Hellmann, Lehmann, Auer, Int. Journal Semantic Web Inf. Syst, 2009

Evaluation Setup

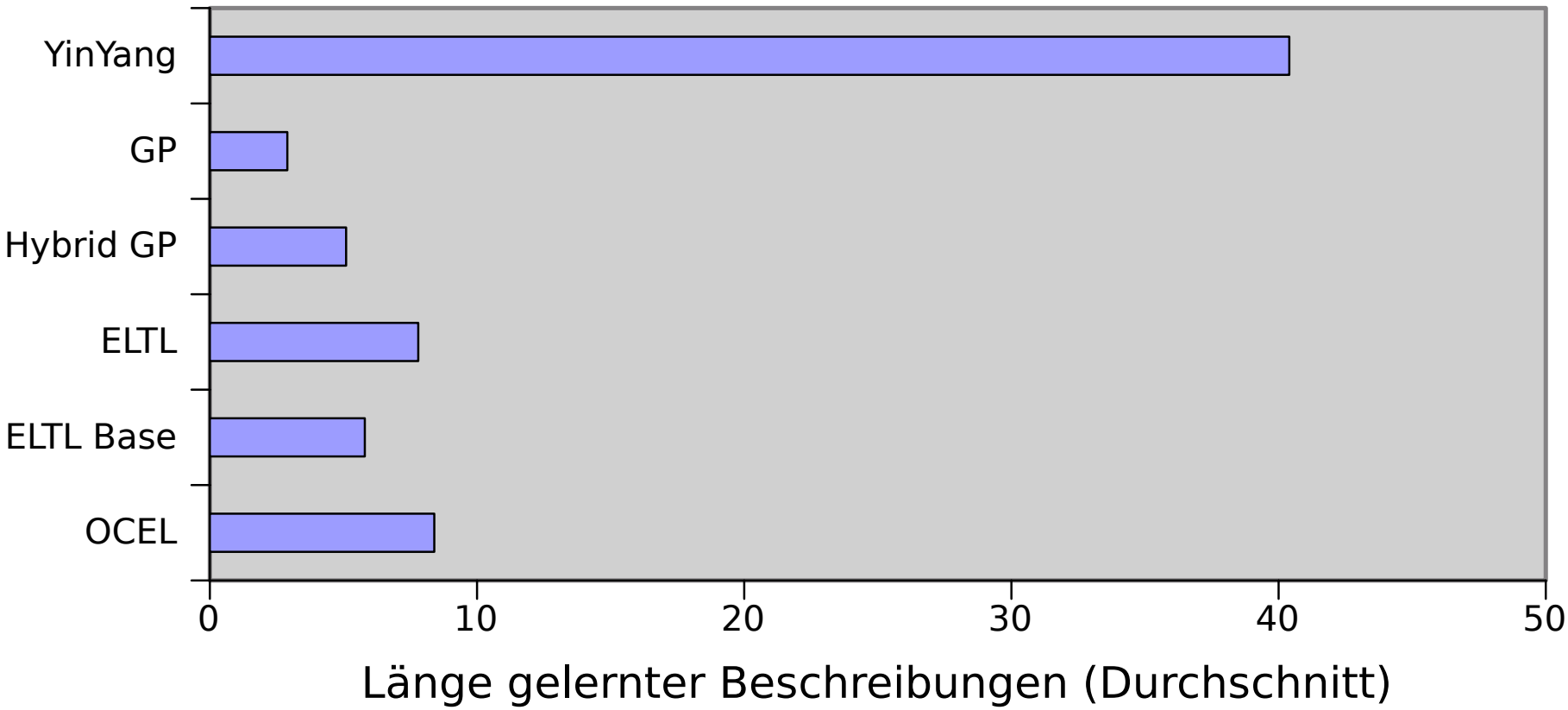
- lack of evaluation standards in OWL/DL learning
- procedure: convert existing benchmarks to OWL (time consuming, requires domain knowledge)
- measure predictive accuracy in ten fold cross validation
- part 1: evaluation against other OWL/DL learning systems
- part 2: evaluation against other ML systems (carcinogenesis problem)
- part 3: evaluation of ontology enginering

Evaluation: Accuracy



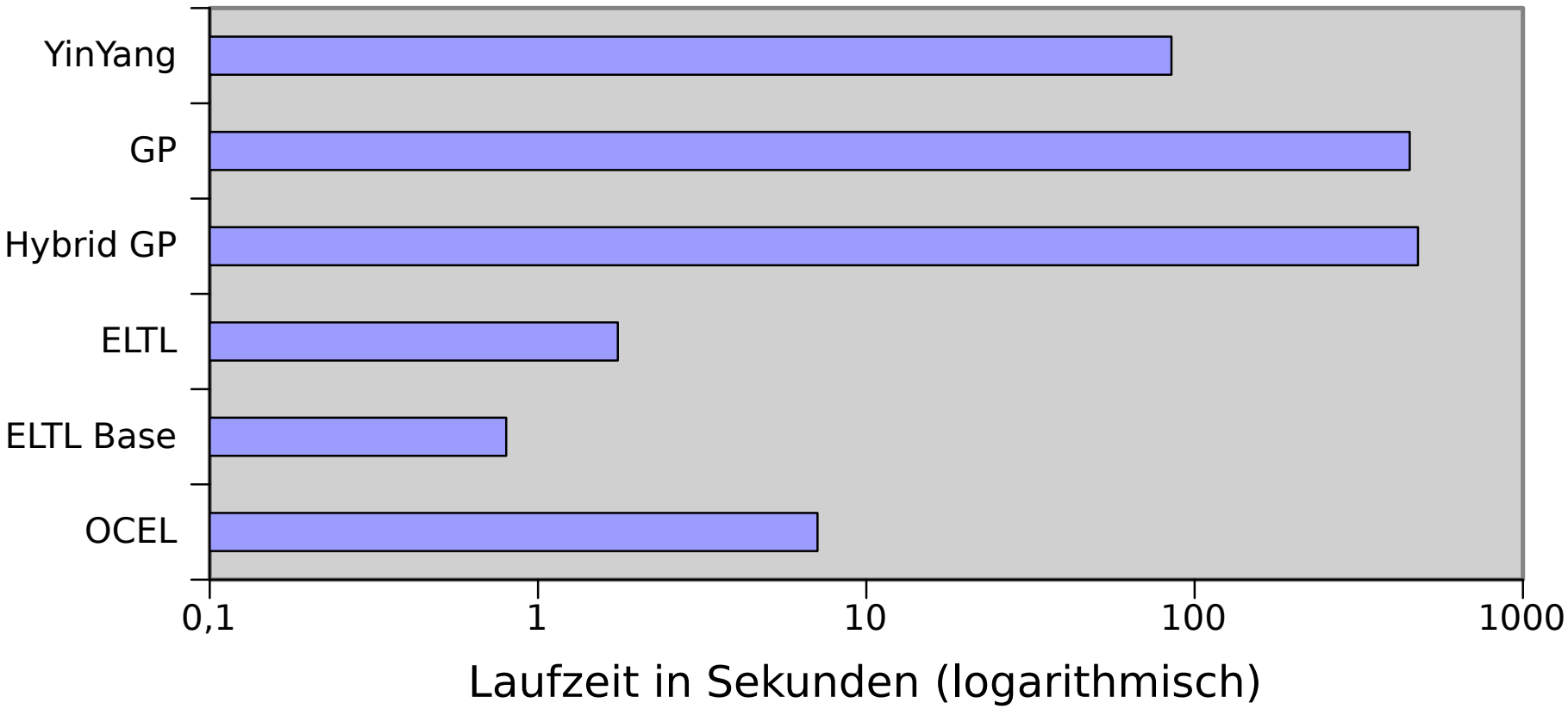
- Collection of 6 Benchmarks
- OCEL often stat. significantly better than other algorithms for most benchmarks

Evaluation: Readability



- YinYang generates significantly longer solutions

Evaluation: Runtime

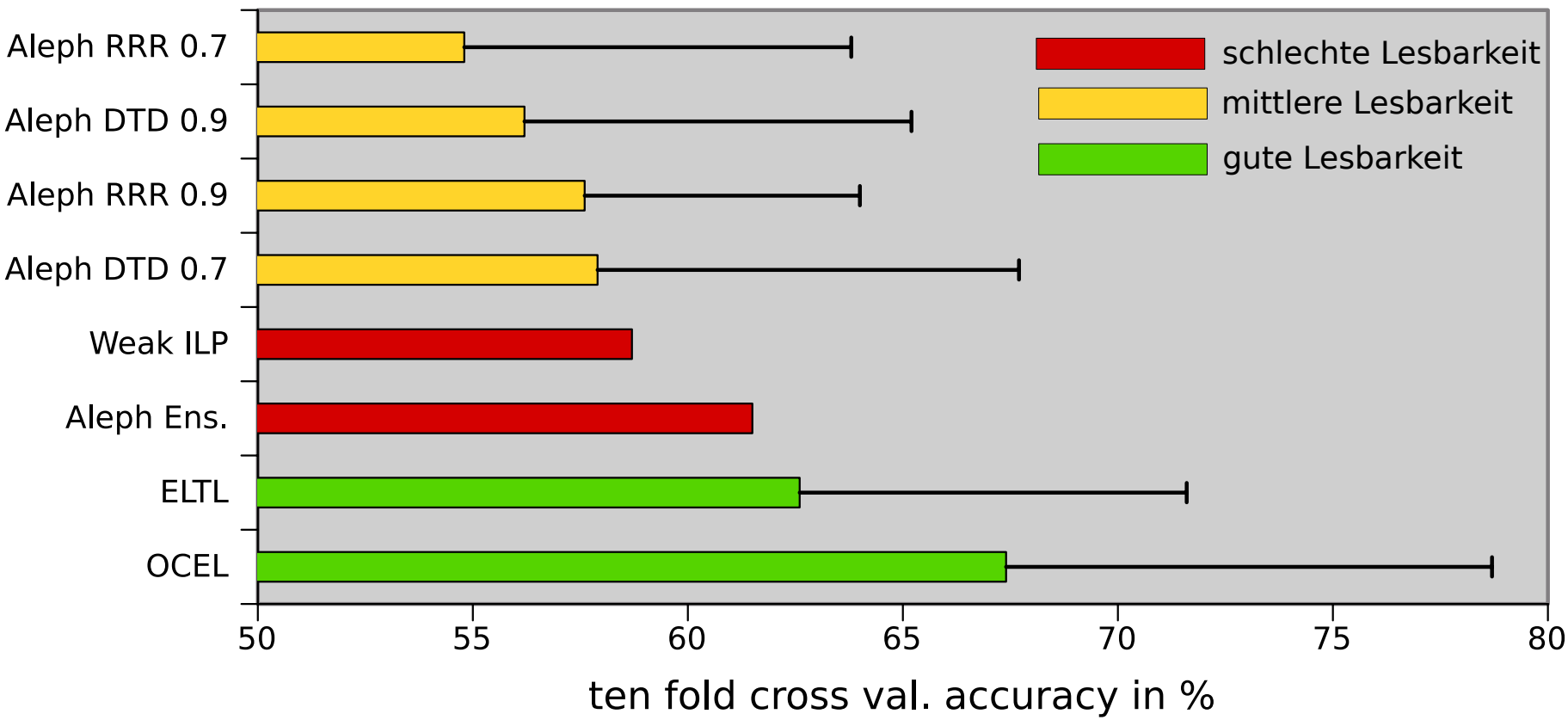


Carcinogenesis

- goal: predict whether chemical compounds cause cancer
- Why?
  - more than 1000 new substances each year
  - substances can often only be tested via long and expensive experiments on rats and mice
- background knowledge:
  - database of US National Toxicology Program (NTP)
  - converted from Prolog to OWL

"Obtaining accurate structural alerts for the causes of chemical cancers is a problem of great scientific and humanitarian value." (A. Srinivasan, R.D. King, S.H. Muggleton, M.J.E. Sternberg 1997)

Carcinogenesis



- very challenging problem: low accuracy, high standard deviation
- OCEL stat. sign. better than most other approaches

Ontology Learning Evaluation

- 5 PhD studens
- 5 real ontologies in different domains
- 998 decision of each test person for 92 classes
- in 35% of the cases accepted suggestions for ontology enhancements
- problem: ontology quality, modelling errors (unsatisfiable classes, disjunction and conjunction confused etc.)

DL-Learner Project

- DL-Learner Open-Source-Projekt: `http://dl-learner.org`, `http://sf.net/projects/dl-learner`
- extensible platform for different learning problems and algorithms
- Interfaces: command line, GUI, Web-Service
- supports common OWL formats
- allows different reasoners (via OWL API, DIG, OWLLink)
- sourceforge.net (Open Source Portal): 4000 Downloads
- mloss.org (ML & Open Source Software): 1600 Downloads

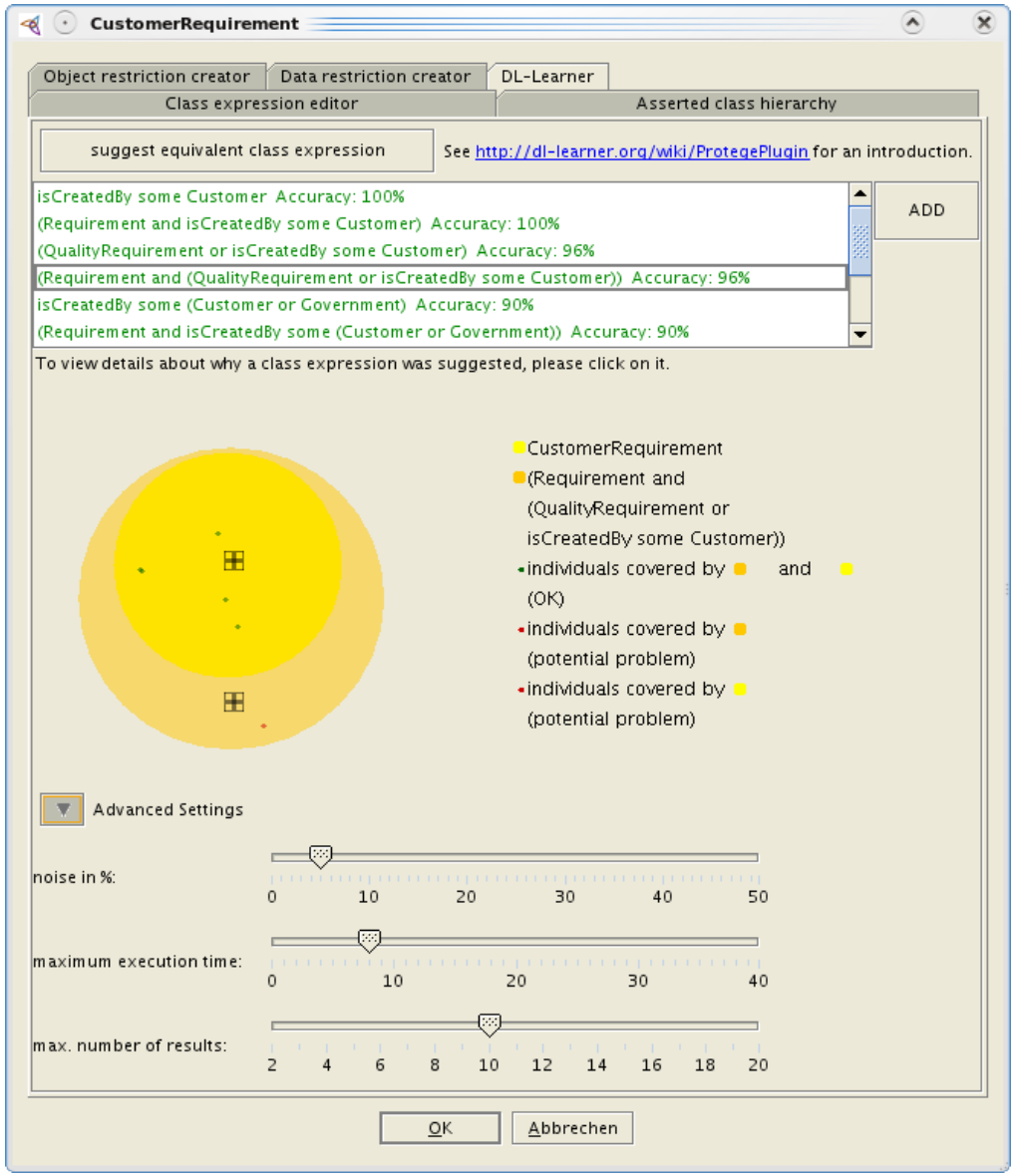
"DL-Learner: Learning Concepts in Description Logics",
Jens Lehmann, Journal of Machine Learning Research (JMLR), 2009

Applications

- “classical“ ML problems
  - carcinogenesis
  - other biomedical tasks

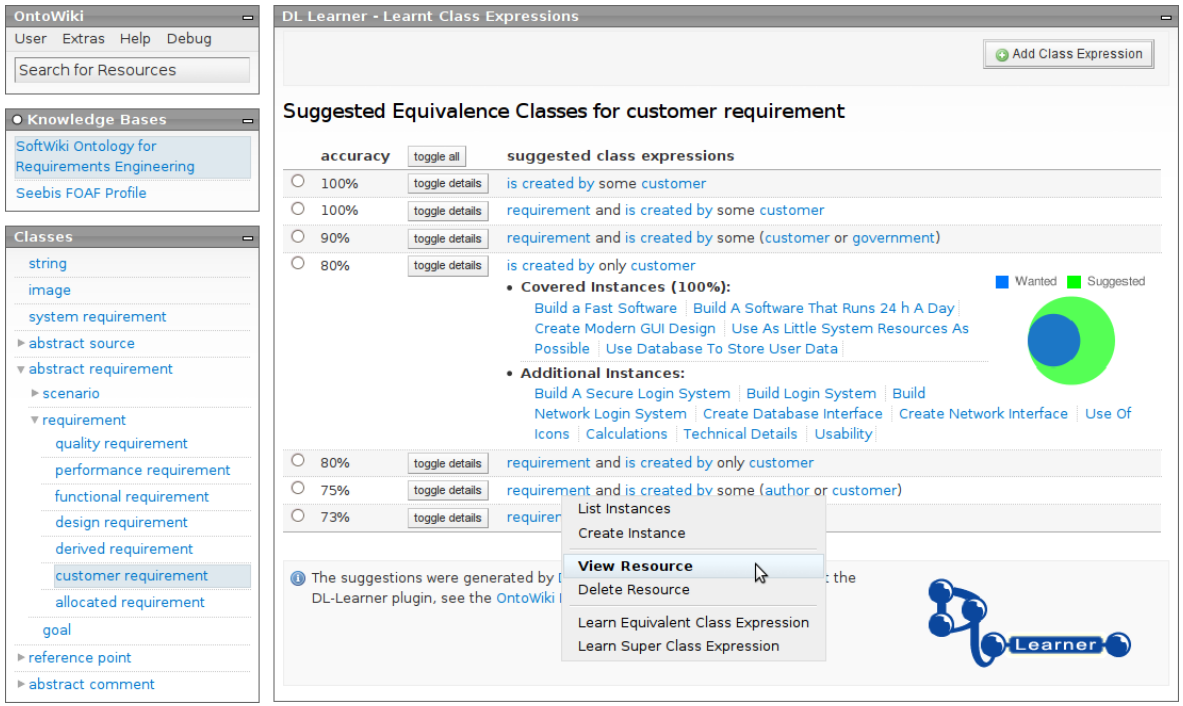
Applications

- "classical" ML problems
  - carcinogenesis
  - other biomedical tasks
- Ontology Learning
  - Protégé Plugin



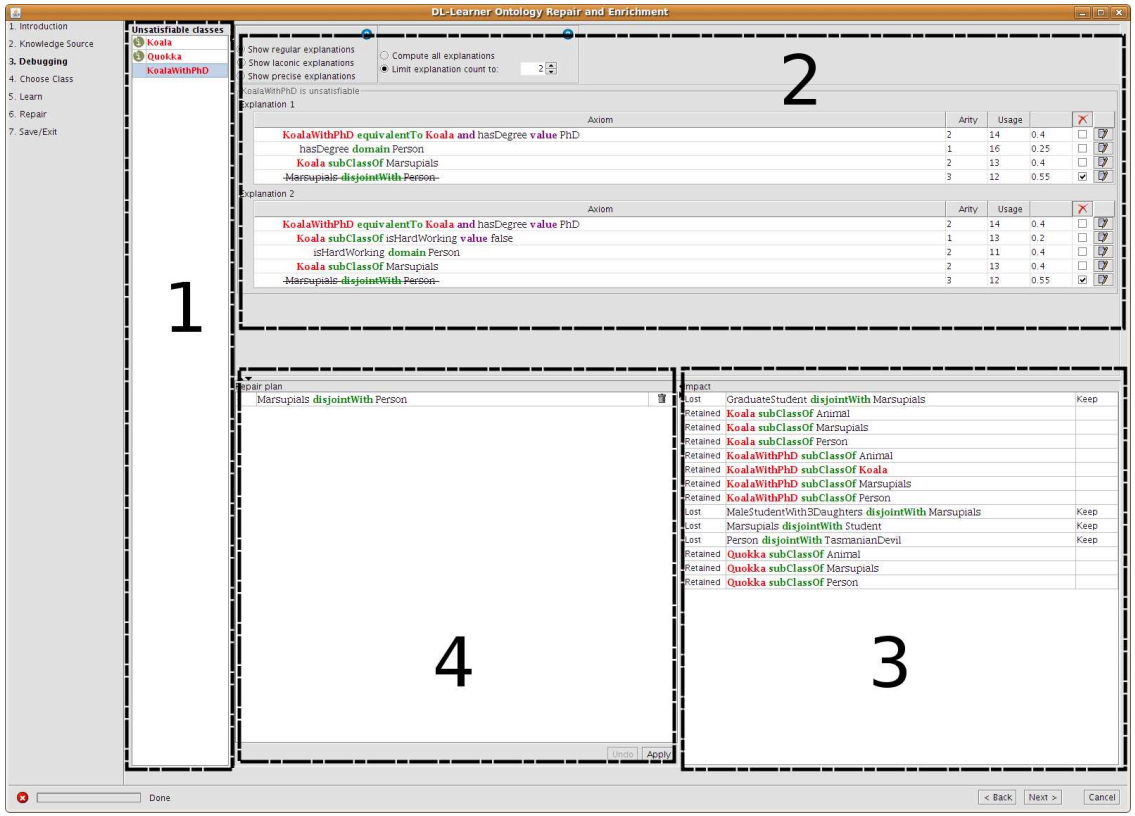
Applications

- "classical" ML problems
  - carcinogenesis
  - other biomedical tasks
- Ontology Learning
  - Protégé Plugin
  - OntoWiki Plugin



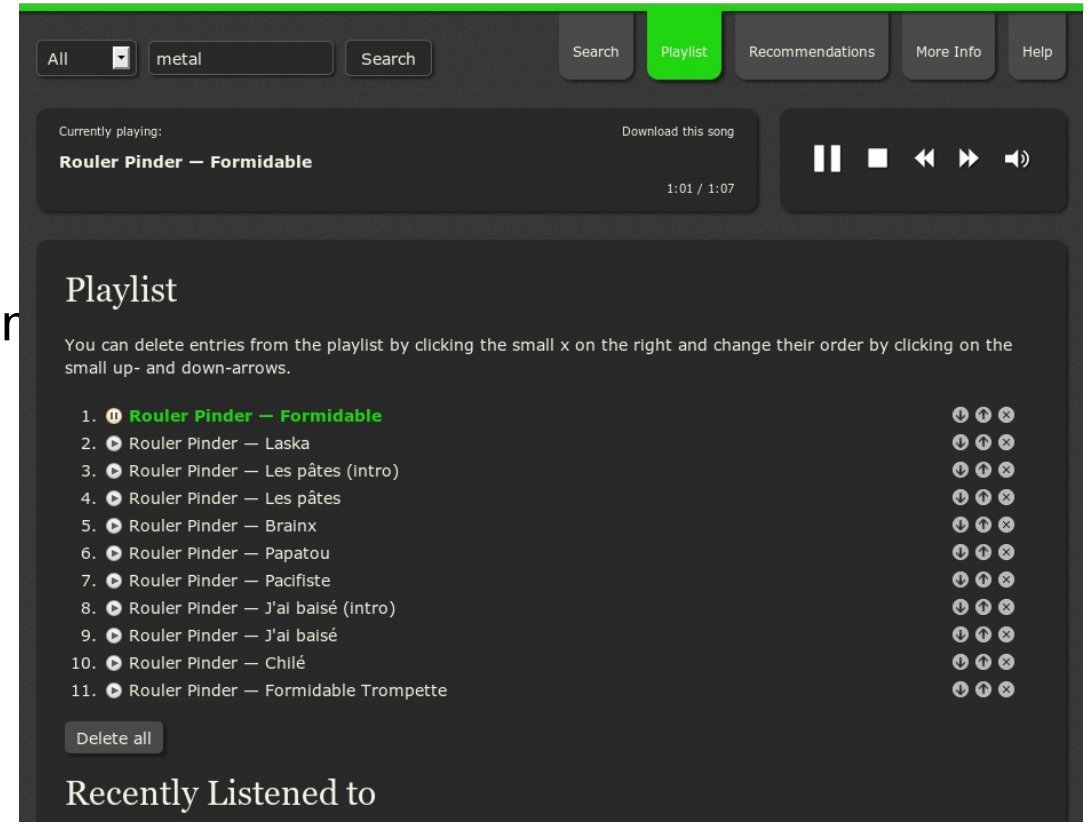
Applications

- "classical" ML problems
  - carcinogenesis
  - other biomedical tasks
- Ontology Learning
  - Protégé Plugin
  - OntoWiki Plugin
  - ORE



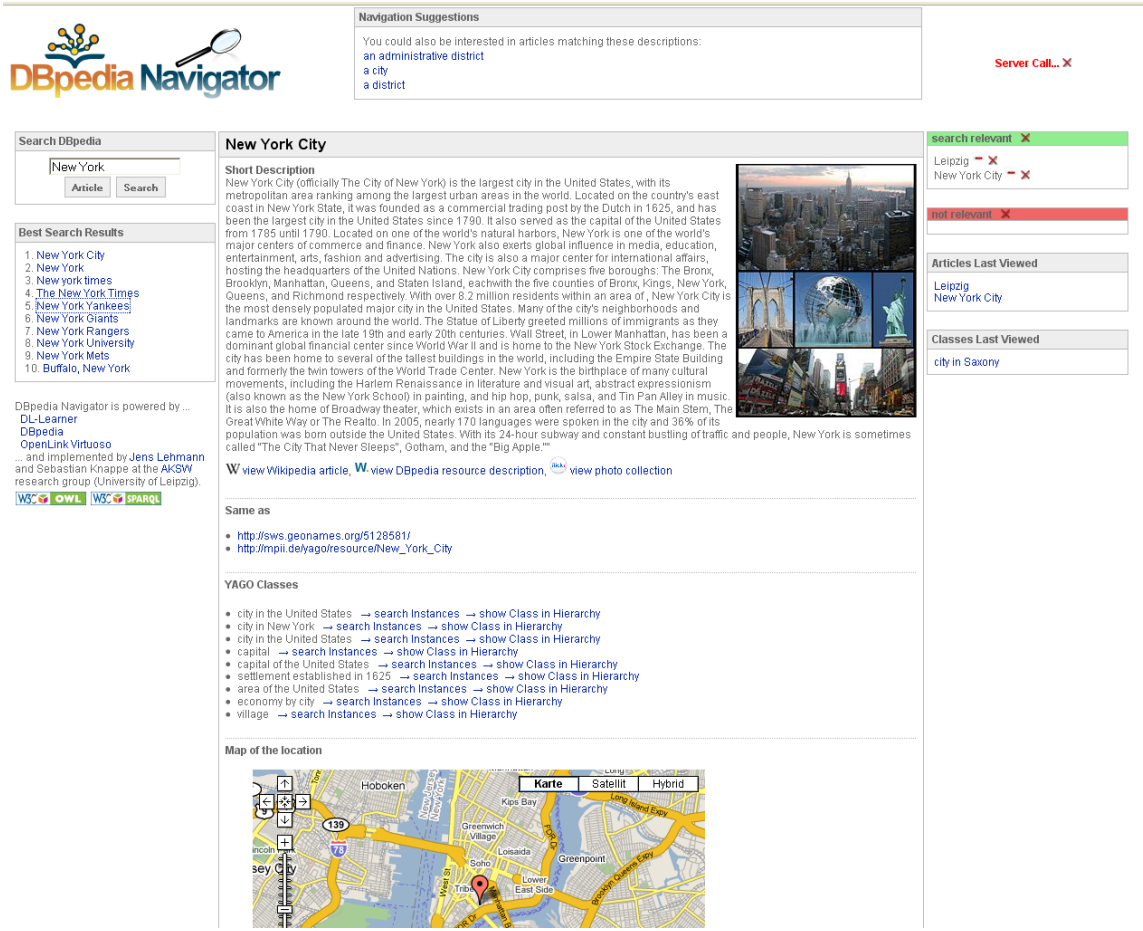
Applications

- "classical" ML problems
  - carcinogenesis
  - other biomedical tasks
- Ontology Learning
  - Protégé Plugin
  - OntoWiki Plugin
  - ORE
- Recommendation/Navigation
  - moosique.net



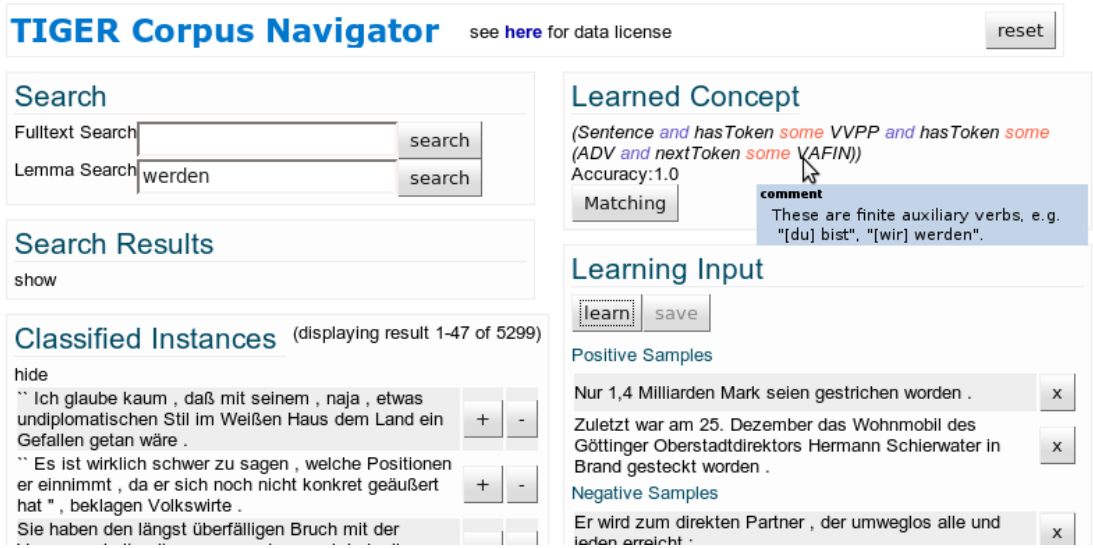
Applications

- "classical" ML problems
  - carcinogenesis
  - other biomedical tasks
- Ontology Learning
  - Protégé Plugin
  - OntoWiki Plugin
  - ORE
- Recommendation/Navigation
  - moosique.net
  - DBpedia Navigator



Applications

- "classical" ML problems
  - carcinogenesis
  - other biomedical tasks
- Ontology Learning
  - Protégé Plugin
  - OntoWiki Plugin
  - ORE
- Recommendation/Navigation
  - moosique.net
  - DBpedia Navigator
- other/external:
  - ISS (Gerken et al.)
  - Learning in Probabilistic DLs (Ochoa Luna et al.)
  - TIGER Corpus Navigator (Hellmann et al.)



Outline

Conclusions

- Ontology Learning is a diverse research area involving several research disciplines (NLP, Machine Learning, Ontology Engineering)
- approaches vary in used data sources and the expressiveness of the created ontologies
- refinement operator based learning as one method for learning definitions (with applications outside of learning ontologies)
- new Wiki (under construction): `http://ontology-learning.net`
- new ontology learning book in 2011