Rule Induction and Reasoning in Knowledge Graphs

Daria Stepanova

Bosch Center for Artificial Intelligence, Renningen, Germany

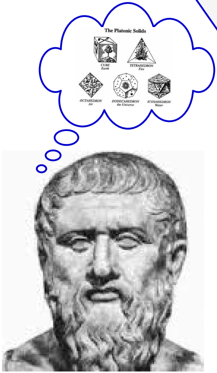
ODSC 2019, 21.11.2019

What is Knowledge?

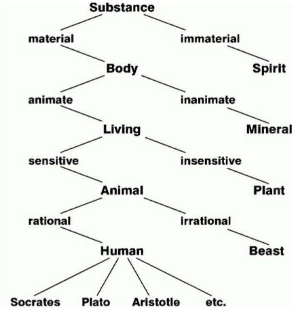
Plato: *"Knowledge is justified true belief"*

350 BC

Personal



External



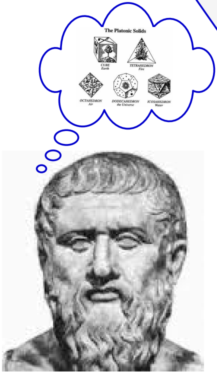
Tree of Knowledge

What is Knowledge?

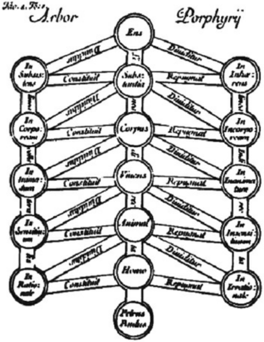
Plato: *"Knowledge is justified true belief"*

350 BC

Personal



External

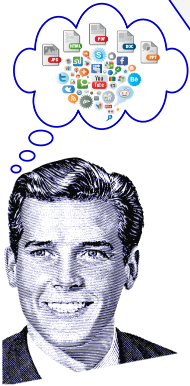


Knowledge Graphs as Digital Knowledge

"Digital knowledge is semantically enriched machine processable data"

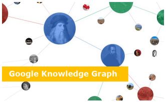
2019

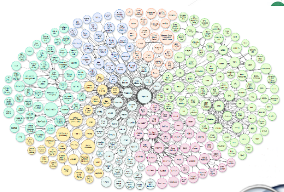
Personal



External

yago
select knowledge





WIKIDATA

NELL

Knowledge Graph

Semantic Web Search

Google | winner of Australian Open 2018

Roger Federer

Tennis player

rogerfederer.com

Roger Federer is a Swiss professional tennis player who is currently ranked world No. 10 by the Association of Tennis Professionals. Many players and analysts have called him the greatest tennis player of all time. Wikipedia

Born: August 8, 1981 (age 35 years), Basel, Switzerland

Height: 1.85 m

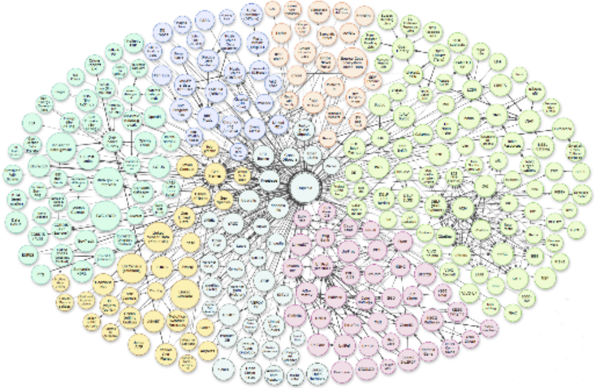
Weight: 85 kg

Spouse: Mirka Federer (m. 2009)

Children: Lenny Federer, Myla Rose Federer, Charlene Riva Federer, Leo Federer

Semantic Web Search

$\exists X\ winnerOf(X, AustralianOpen2018)$



Roger Federer

Tennis player

rogerfederer.com

Roger Federer is a Swiss professional tennis player who is currently ranked world No. 10 by the Association of Tennis Professionals. Many players and analysts have called him the greatest tennis player of all time. Wikipedia

Born: August 8, 1981 (age 35 years), Basel, Switzerland

Height: 1.85 m

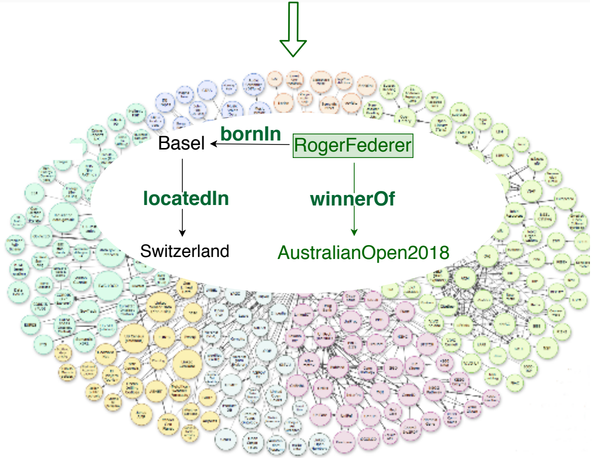
Weight: 85 kg

Spouse: Mirka Federer (m. 2009)

Children: Lenny Federer, Myla Rose Federer, Charlene Riva Federer, Leo Federer

Semantic Web Search

Google winner of Australian Open 2018



Roger Federer

Tennis player

rogerfederer.com

Roger Federer is a Swiss professional tennis player who is currently ranked world No. 10 by the Association of Tennis Professionals. Many players and analysts have called him the greatest tennis player of all time. Wikipedia

Born: August 8, 1981 (age 35 years), Basel, Switzerland

Height: 1.85 m

Weight: 85 kg

Spouse: Mirka Federer (m. 2009)

Children: Lenny Federer, Myla Rose Federer, Charlene Riva Federer, Leo Federer

Semantic Web Search

living place of the winner of australian open 2018

All News Images Videos Maps More Settings Tools

About 1,220,000,000 results (1.10 seconds)

2018 Australian Open - Wikipedia
https://en.wikipedia.org/wiki/2018_Australian_Open ▾
Roger Federer was the defending **champion** in the men's singles event and successfully retained his title (his sixth), defeating Marin Čilić in the final, while Caroline Wozniacki **won** the women's title, defeating Simona Halep in the final.

Venue: Melbourne Park
Location: Melbourne, Victoria, Australia
Prize money: A$55,000,000
Draw: 128S / 64D /

Missing: ~~living~~ | Must include: living

Semantic Web Search

wife of Roger Federer

All Images News Videos Maps More Settings Tools

About 42,200,000 results (0.50 seconds)

Roger Federer / Wife

Mirka Federer

m. 2009

Miroslava "Mirka" Federer is a Slovak-born Swiss former professional tennis player. She reached her career-high WTA singles ranking of world No. 76 on 10 September 2001 and a doubles ranking of No. 215 on 24 August 1998. She is the wife of tennis player Roger Federer, having first met him at the 2000 Summer Olympics. Wikipedia

Semantic Web Search

living place of Mirka Federer

All Images News Shopping Videos More Settings Tools

About 1.910.000 results (0,92 seconds)

Mirka Federer / **Residence**

Bottmingen, Switzerland

Human Reasoning

$livesIn(Y,Z) \leftarrow marriedTo(X,Y),$ $livesIn(X,Z)$	*Married people live together*
$marriedTo(mirka, roger)$	*Mirka is married to Roger*
$livesIn(mirka, bottmingen)$	*Mirka lives in Bottmingen*
___	___

Human Reasoning

$$\mathit{livesIn}(Y, Z) \leftarrow \mathit{marriedTo}(X, Y), \mathit{livesIn}(X, Z)$$

Married people live together

$\mathit{marriedTo}(\mathit{mirka}, \mathit{roger})$

Mirka is married to Roger

$\mathit{livesIn}(\mathit{mirka}, \mathit{bottmingen})$

Mirka lives in Bottmingen

$\mathit{livesIn}(\mathit{roger}, \mathit{bottmingen})$

Roger lives in Bottmingen

livesIn →

Human Reasoning

$$\mathit{livesIn}(Y, Z) \leftarrow \mathit{marriedTo}(X, Y),\ \mathit{livesIn}(X, Z)$$

Married people live together

$\mathit{marriedTo}(\mathit{mirka}, \mathit{roger})$

Mirka is married to Roger

$\mathit{livesIn}(\mathit{mirka}, \mathit{bottmingen})$

Mirka lives in Bottmingen

$\mathit{livesIn}(\mathit{roger}, \mathit{bottmingen})$

Roger lives in Bottmingen

livesIn →

But where can a machine get such rules from?

Applications of Rule Learning

- Fact prediction
- Fact checking
- Data cleaning
- Domain description
- Finding trends in KGs . . .

Horn Rules

Rule: $\underbrace{a}_{\text{head}} \leftarrow \underbrace{b_1, \ldots, b_m.}_{\text{body}}$

Informal semantics: If $b_1, \ldots, b_m$ are true, then a must be true.

Logic program: Set of rules

Example: ground rule

% If Mirka is married to Roger and lives in B., then Roger lives there too
$livesIn(roger, bottmingen) \leftarrow isMarried(mirka, roger), livesIn(mirka, bottmingen)$

Horn Rules

Rule: $\underbrace{a}_{\text{head}} \leftarrow \underbrace{b_1, \ldots, b_m.}_{\text{body}}$

Informal semantics: If $b_1, \ldots, b_m$ are true, then a must be true.

Logic program: Set of rules

Example: non-ground rule

% Married people live together
$livesIn(Y, Z) \leftarrow isMarried(X, Y), livesIn(X, Z)$

Nonmonotonic Rules

Rule: $\underbrace{a}_{\text{head}} \leftarrow \underbrace{b_1, \ldots, b_m, \; not \; b_{m+1}, \ldots, \; not \; b_n.}_{\text{body}}$

Informal semantics: If $b_1, \ldots, b_m$ are true and none of $b_{m+1}, \ldots, b_n$ is known, then a must be true.

Closed World Assumption (CWA): facts not known to be true are false

Example: nonmonotonic rule

% Two married live together unless one is a researcher
$livesIn(Y, Z) \leftarrow isMarried(X, Y), livesIn(X, Z), not \;\; researcher(Y)$

Nonmonotonic Rules

Rule: $\underbrace{a}_{\text{head}} \leftarrow \underbrace{b_1, \ldots, b_m, \ \textit{not}\ b_{m+1}, \ldots, \ \textit{not}\ b_n.}_{\text{body}}$

Informal semantics: If $b_1, \ldots, b_m$ are true and none of $b_{m+1}, \ldots, b_n$ is known, then a must be true.

Closed World Assumption (CWA): facts not known to be true are false

not is different from $\neg$!

% At a rail road crossing cross the road if **no train is known** to approach"
$walk \leftarrow at(L), crossing(L), not\ \ train\_approaches(L)$

% At a rail road crossing cross the road if **no train** approaches
$walk \leftarrow at(L), crossing(L), \neg train\_approaches(L)$

Answer Set Programs

Evaluation of ASP programs is model-based

Answer set program (ASP) is a set of nonmonotonic rules

(1) $isMarriedTo(mary, john)$ (2) $livesIn(mary, ulm)$

$$(3)\ livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), livesIn(X, Z),\ not\ researcher(Y)$$

Answer Set Programs

Evaluation of ASP programs is model-based

1. Grounding: substitute all variables with constants in all possible ways

Answer set program (ASP) is a set of nonmonotonic rules

(1) $isMarriedTo(mary, john)$ (2) $livesIn(mary, ulm)$

(3) $livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), livesIn(X, Z),$
$not\ researcher(Y)$

Answer Set Programs

Evaluation of ASP programs is model-based

1. Grounding: substitute all variables with constants in all possible ways

Answer set program (ASP) is a set of nonmonotonic rules

(1) $isMarriedTo(mary, john)$ (2) $livesIn(mary, ulm)$

(3) $livesIn(john, ulm) \leftarrow isMarriedTo(mary, john), livesIn(mary, ulm),$
$not\ researcher(john)$

Answer Set Programs

Evaluation of ASP programs is model-based

1. Grounding: substitute all variables with constants in all possible ways
2. Solving: compute a minimal model (answer set) I satisfying all rules

Answer set program (ASP) is a set of nonmonotonic rules

(1) $isMarriedTo(mary, john)$ (2) $livesIn(mary, ulm)$

(3) $livesIn(john, ulm) \leftarrow isMarriedTo(mary, john), livesIn(mary, ulm),$
$not\ researcher(john)$

$I = \{isMarriedTo(mary, john), livesIn(mary, ulm), livesIn(john, ulm)\}$

CWA: $researcher(john)$ can not be derived, thus it is false

Answer Set Programs

Evaluation of ASP programs is model-based

1. Grounding: substitute all variables with constants in all possible ways
2. Solving: compute a minimal model (answer set) I satisfying all rules

Answer set program (ASP) is a set of nonmonotonic rules

$$(1)\ isMarriedTo(mary, john) \quad (2)\ livesIn(mary, ulm)$$

$$(3)\ livesIn(john, ulm) \leftarrow isMarriedTo(mary, john), livesIn(mary, ulm), \\ not\ researcher(john)$$

$$(4)\ researcher(john)$$

$$I = \{isMarriedTo(mary, john), livesIn(mary, ulm), \overset{researcher(john)}{\cancel{livesIn(john, ulm)}}\}$$

Answer Set Programs

Evaluation of ASP programs is model-based

1. Grounding: substitute all variables with constants in all possible ways
2. Solving: compute a minimal model (answer set) I satisfying all rules

Answer set program (ASP) is a set of nonmonotonic rules

(1) $isMarriedTo(mary, john)$ (2) $livesIn(mary, ulm)$

(3) $livesIn(john, ulm) \leftarrow isMarriedTo(mary, john), livesIn(mary, ulm),$
$not\ researcher(john)$

(4) $researcher(john)$

$$I = \{isMarriedTo(mary, john), livesIn(mary, ulm), \overset{researcher(john)}{\cancel{livesIn(john, ulm)}}\}$$

Particularly suited for reasoning under incompleteness!

Reasoning with Incomplete Information

Default Reasoning

Assume normal state of affairs, unless there is evidence to the contrary

By default married people live together.

Reasoning with Incomplete Information

Default Reasoning

Assume normal state of affairs, unless there is evidence to the contrary

By default married people live together.

Abduction

Choose between several explanations that explain an observation

John and Mary live together. They must be married.

Reasoning with Incomplete Information

Default Reasoning

Assume normal state of affairs, unless there is evidence to the contrary

By default married people live together.

Abduction

Choose between several explanations that explain an observation

John and Mary live together. They must be married.

Induction

Generalize a number of similar observations into a hypothesis

Given many examples of spouses living together generalize this knowledge.

Reasoning with Incomplete Information

Default Reasoning

Assume normal state of affairs, unless there is evidence to the contrary

By default married people live together.

Abduction

Choose between several explanations that explain an observation

John and Mary live together. They must be married.

Induction

Generalize a number of similar observations into a hypothesis

Given many examples of spouses living together generalize this knowledge.

History of Inductive Learning

- AI & Machine Learning 1960s-70s:
  Banerji, Plotkin, Vere, Michalski, ...
- AI & Machine Learning 1980s:
  Shapiro, Sammut, Muggleton, ...
- Inductive Logic Programming (ILP) 1990s:
  Muggleton, Quinlan, De Raedt, ...
- Statistical Relational Learning 2000s:
  Getoor, Koller, Domingos, Sato, ...

Learning from Examples

Inductive Learning from Examples [Muggleton, 1991]

Given:

- $E^+ = \{fatherOf(john, mary), fatherOf(david, steve)\}$
- $E^- = \{fatherOf(kathy, ellen), fatherOf(john, steve)\}$
- $T = \{parentOf(john, mary), male(john),$
  $parentOf(david, steeve), male(david),$
  $parentOf(kathy, ellen), female(kathy)\}$
- Language bias: Horn rules with 2 body atoms

Learning from Examples

Inductive Learning from Examples [Muggleton, 1991]

Given:

- $E^+ = \{fatherOf(john, mary), fatherOf(david, steve)\}$
- $E^- = \{fatherOf(kathy, ellen), fatherOf(john, steve)\}$
- $T = \{parentOf(john, mary), male(john),$
  $parentOf(david, steeve), male(david),$
  $parentOf(kathy, ellen), female(kathy)\}$
- Language bias: Horn rules with 2 body atoms

Possible hypothesis:

- $Hyp: \ fatherOf(X, Y) \leftarrow parentOf(X, Y), male(X)$

Learning from Interpretations

Inductive Learning from Interpretations [Raedt and Dzeroski, 1994]

Given:

- $I = \{isMarriedTo(mirka, roger), livesIn(mirka, b),$
  $livesIn(roger, b), bornIn(mirka, b)\}$
- $T = \{isMarriedTo(mirka, roger);\ bornIn(mirka, b);$
  $livesIn(X, Y) \leftarrow bornIn(X, Y)\}$
- Language bias: Horn rules with 2 body atoms

Learning from Interpretations

Inductive Learning from Interpretations [Raedt and Dzeroski, 1994]

Given:

- $I = \{isMarriedTo(mirka, roger), livesIn(mirka, b),$
  $livesIn(roger, b), bornIn(mirka, b)\}$
- $T = \{isMarriedTo(mirka, roger);\ bornIn(mirka, b);$
  $livesIn(X, Y) \leftarrow bornIn(X, Y)\}$
- Language bias: Horn rules with 2 body atoms

Possible Hypothesis:

- $Hyp : livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), bornIn(X, Z)$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger),\ \ \theta = \{X/roger\}$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger), \;\; \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger), \ \ \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, bosch), researcher(Z)\},$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger), \ \ \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, bosch), researcher(Z)\}$,
    $\theta = \{X/Z, Y/bosch\}$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger), \;\; \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, bosch), researcher(Z)\}$,
    $\theta = \{X/Z, Y/bosch\}$
- Generalization as entailment
  - Logic program: $Hyp1 \succeq Hyp2$ iff $Hyp1 \models Hyp2$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger), \;\; \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, bosch), researcher(Z)\},$
    $\theta = \{X/Z, Y/bosch\}$
- Generalization as entailment
  - Logic program: $Hyp1 \succeq Hyp2$ iff $Hyp1 \models Hyp2$
    $\underbrace{person(X) \leftarrow researcher(X)}_{Hyp1} \; \underbrace{person(mat) \leftarrow researcher(mat)}_{Hyp_2}$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
    - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
      $person(X) \succeq person(roger), \;\; \theta = \{X/roger\}$
    - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$
      $\{worksAt(X,Y)\} \succeq \{worksAt(Z,bosch), researcher(Z)\},$
      $\theta = \{X/Z, Y/bosch\}$
- Generalization as entailment
    - Logic program: $Hyp1 \succeq Hyp2$ iff $Hyp1 \models Hyp2$
      $\underbrace{person(X) \leftarrow researcher(X)}_{Hyp1} \; \underbrace{person(mat) \leftarrow researcher(mat)}_{Hyp_2}$

$$Hyp1 \succeq Hyp2$$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger), \;\; \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, bosch), researcher(Z)\},$
    $\theta = \{X/Z, Y/bosch\}$
- Generalization as entailment
  - Logic program: $Hyp1 \succeq Hyp2$ iff $Hyp1 \models Hyp2$
    $\underbrace{person(X) \leftarrow researcher(X)}_{Hyp1} \; \underbrace{person(X) \leftarrow researcher(X), alive(X)}_{Hyp2}$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger), \;\; \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, bosch), researcher(Z)\}$,
    $\theta = \{X/Z, Y/bosch\}$
- Generalization as entailment
  - Logic program: $Hyp1 \succeq Hyp2$ iff $Hyp1 \models Hyp2$
    $\underbrace{person(X) \leftarrow researcher(X)}_{Hyp1} \; \underbrace{person(X) \leftarrow researcher(X), alive(X)}_{Hyp2}$

$$Hyp1 \succeq Hyp2$$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger), \;\; \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, bosch), researcher(Z)\},$
    $\theta = \{X/Z, Y/bosch\}$
- Generalization as entailment
  - Logic program: $Hyp1 \succeq Hyp2$ iff $Hyp1 \models Hyp2$
    $\underbrace{person(X) \leftarrow researcher(X)}_{Hyp1} \; \underbrace{person(X) \leftarrow researcher(X), alive(X)}_{Hyp2}$
    $Hyp1 \succeq Hyp2$
  - Relative entailment: $Hyp1 \succeq Hyp2$ wrt T iff $Hyp1 \cup T \models Hyp2$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger), \;\; \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, bosch), researcher(Z)\},$
    $\theta = \{X/Z, Y/bosch\}$
- Generalization as entailment
  - Logic program: $Hyp1 \succeq Hyp2$ iff $Hyp1 \models Hyp2$
    $\underbrace{person(X) \leftarrow researcher(X)}_{Hyp1} \; \underbrace{person(X) \leftarrow researcher(X), alive(X)}_{Hyp2}$
    $Hyp1 \succeq Hyp2$
  - Relative entailment: $Hyp1 \succeq Hyp2$ wrt T iff $Hyp1 \cup T \models Hyp2$
    $livesIn(roger, bottmingen)$? $livesIn(roger, switzerland)$

Common Techniques in ILP

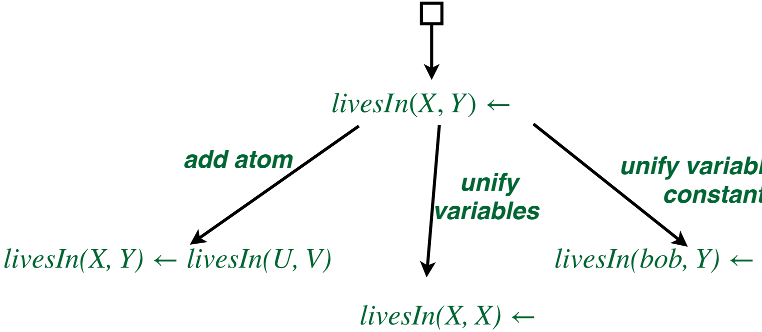
- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger),\ \ \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, bosch), researcher(Z)\}$,
    $\theta = \{X/Z, Y/bosch\}$
- Generalization as entailment
  - Logic program: $Hyp1 \succeq Hyp2$ iff $Hyp1 \models Hyp2$
    $\underbrace{person(X) \leftarrow researcher(X)}_{Hyp1}\ \underbrace{person(X) \leftarrow researcher(X), alive(X)}_{Hyp2}$
    $Hyp1 \succeq Hyp2$
  - Relative entailment: $Hyp1 \succeq Hyp2$ wrt T iff $Hyp1 \cup T \models Hyp2$
    $livesIn(roger, bottmingen)$? $livesIn(roger, switzerland)$
    $T:\ livesIn(X, switzerland) \leftarrow livesIn(X, bottmingen)$

Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger),\ \ \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, bosch), researcher(Z)\},$
    $\theta = \{X/Z, Y/bosch\}$
- Generalization as entailment
  - Logic program: $Hyp1 \succeq Hyp2$ iff $Hyp1 \models Hyp2$
    $\underbrace{person(X) \leftarrow researcher(X)}_{Hyp1}\ \underbrace{person(X) \leftarrow researcher(X), alive(X)}_{Hyp2}$
    $Hyp1 \succeq Hyp2$
  - Relative entailment: $Hyp1 \succeq Hyp2$ wrt T iff $Hyp1 \cup T \models Hyp2$
    $livesIn(roger, bottmingen) \succeq livesIn(roger, switzerland)$
    $T:\ livesIn(X, switzerland) \leftarrow livesIn(X, bottmingen)$

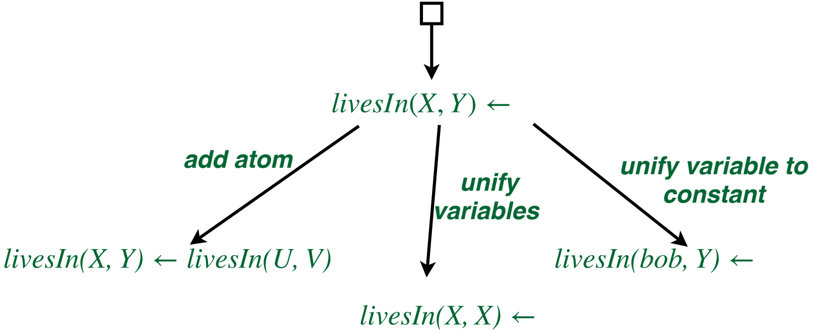
Common Techniques in ILP

- Clause refinement [Shapiro, 1991]: e.g., MIS, FOIL, etc.
  - Explore clause search space from general to specific or vice versa to find a hypothesis that covers all examples.



Common Techniques in ILP

- Clause refinement [Shapiro, 1991]: e.g., MIS, FOIL, etc.
  - Explore clause search space from general to specific or vice versa to find a hypothesis that covers all examples.



- Inverse entailment [Muggleton, 1995]: e.g., Progol, etc.
  - Properties of deduction to make hypothesis search space finite

Zoo of Other ILP Tasks

ILP tasks can be classified along several dimensions:

- type of the data source, e.g., positive/negative examples, interpretations, answer sets [Law *et al.*, 2015]
- type of the output knowledge, e.g., rules, DL ontologies [Lehmann, 2009]
- the way the data is given as input, e.g., all at once, incrementally [Katzouris *et al.*, 2015]
- availability of an oracle, e.g., human in the loop
- quality of the data source, e.g., noisy [Evans and Grefenstette, 2018]
- data (in)completeness, e.g., OWA vs CWA...
- background knowledge, e.g., DL ontology [d'Amato *et al.*, 2016], hybrid theories [Lisi, 2010]

Classical ILP for KGs

ILP Goal

”The goal of ILP is to develop a correct (and complete) algorithm which efficiently computes hypotheses.” [Sakama, 2005]

Knowledge Graphs

But the world knowledge is complex, and this might not always be possible in the context of KGs due to several issues...

Specialities of KGs

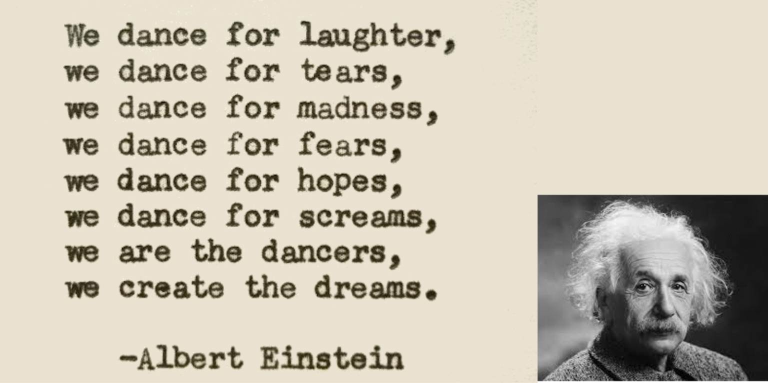
Open World Assumption: negative facts cannot be easily derived

Maybe Roger Federer is a researcher and Albert Einstein was a ballet dancer?

Specialities of KGs

Open World Assumption: negative facts cannot be easily derived

Maybe Roger Federer is a researcher and Albert Einstein was a ballet dancer?



Challenges of Rule Induction from KGs

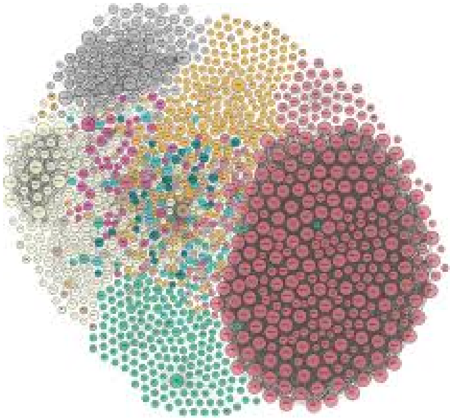
Data bias: KGs are extracted from text, which typically mentions only popular entities and interesting facts about them.

"Man bites dog phenomenon"[1]

[1] https://en.wikipedia.org/wiki/Man_bites_dog_(journalism)

Challenges of Rule Induction from KGs

Huge size: Modern KGs contain billions of facts
E.g., Google KG stores 70 billion facts



Challenges of Rule Induction from KGs

World knowledge is complex, none of its "models" is perfect

Exploratory Data Analysis

Question:

How can we still learn rules from KGs, which do not perfectly fit the data, but still reflect interesting correlations that can predict sufficiently many correct facts?

Answer:

Relational association rule mining! Roots in classical datamining.

Association Rules

- Classical data mining task: Given a transaction database, find out products (called itemsets) that are frequently bought together and form recommendation rules.

Transaction 1	🍎 🍺 🍚 🍗
Transaction 2	🍎 🍺 🍚
Transaction 3	🍎 🍺
Transaction 4	🍎 🍐
Transaction 5	🍼 🍺 🍚 🍗
Transaction 6	🍼 🍺 🍚
Transaction 7	🍼 🍺
Transaction 8	🍼 🍐

Out of 4 people who bought apples, 3 also bought beer.

Some Rule Measures

Support, confidence, lift

Support [🍎] = 4

Transaction 1	🍎 🍺 🍚 🍗
Transaction 2	🍎 🍺 🍚
Transaction 3	🍎 🍺
Transaction 4	🍎 🍐
Transaction 5	🍼 🍺 🍚 🍗
Transaction 6	🍼 🍺 🍚
Transaction 7	🍼 🍺
Transaction 8	🍼 🍐

Some Rule Measures

Support, confidence, lift

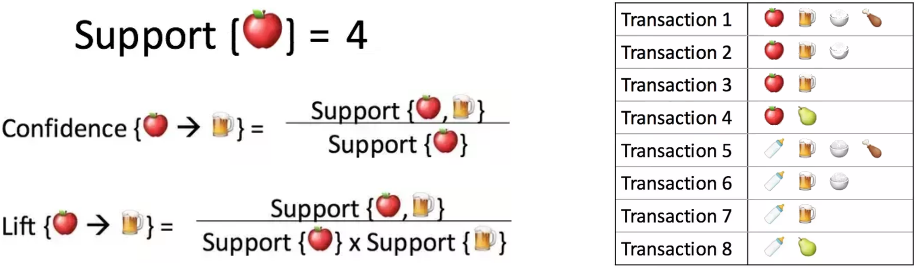
Support [🍎] = 4

$$\text{Confidence}\,\{🍎 \rightarrow 🍺\} = \frac{\text{Support}\,\{🍎, 🍺\}}{\text{Support}\,\{🍎\}}$$

Transaction 1	🍎 🍺 🍚 🍗
Transaction 2	🍎 🍺 🍚
Transaction 3	🍎 🍺
Transaction 4	🍎 🍐
Transaction 5	🍼 🍺 🍚 🍗
Transaction 6	🍼 🍺 🍚
Transaction 7	🍼 🍺
Transaction 8	🍼 🍐

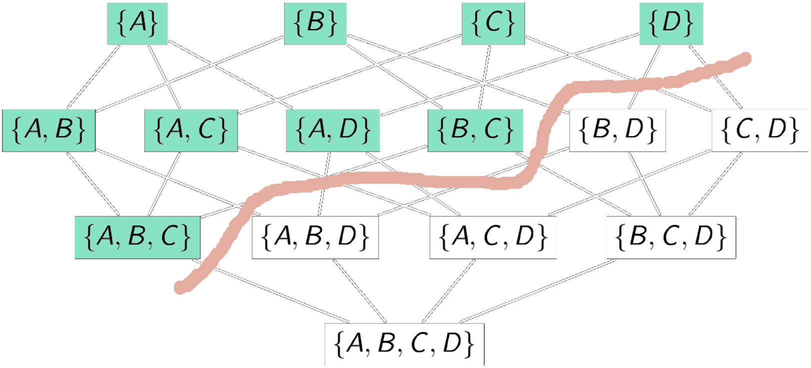
Some Rule Measures

Support, confidence, lift



Frequent Itemset Mining

- A=apple, B=beer... Frequent patterns are in green.
- Monotonicity: any superset of an infrequent pattern is infrequent
  At the heart of Apriori algorithm



Relational Association Rule Learning

- WARMER [Goethals and den Bussche, 2002]
- Upgrade frequent itemsets to frequent conjunctive queries

CQ: return all people with their spouses and living places

$$q_1(X, Y, Z) :- isMarriedTo(X, Y) \wedge livesIn(X, Z)$$

Output: 6 tuples, i.e., $supp(q_1) = 6$

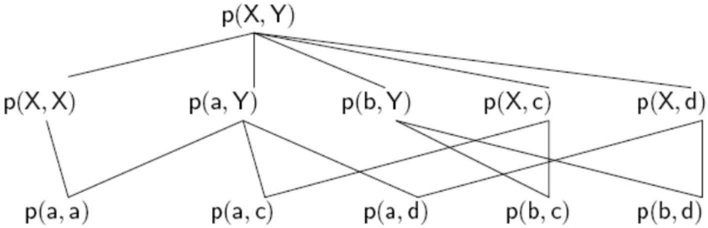
CQ: return all people with their spouses and living places

$$q_2(X, Y, Z) :- isMarriedTo(X, Y) \wedge livesIn(X, Z) \wedge livesIn(Y, Z)$$

Output: 3 tuples, i.e., $supp(q_2) = 3$

Relational Association Rule Learning

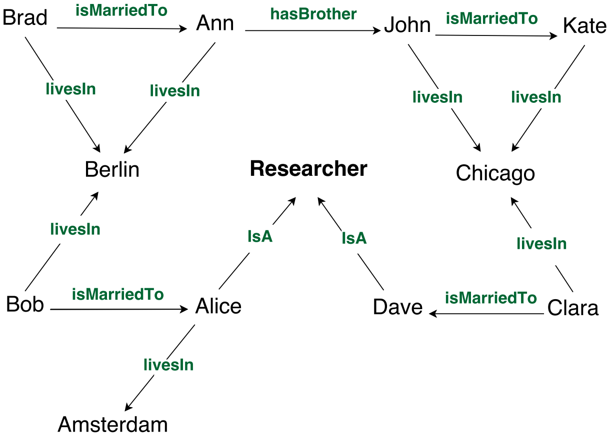
- WARMER [Goethals and den Bussche, 2002]
- Upgrade frequent itemsets to frequent conjunctive queries
  - traverse the lattice
  - get frequent CQs based on user-specified value
  - split into body and head
  - rank based on a rule measure, e.g., confidence



Horn Rule Learning from KGs

WARMER: confidence

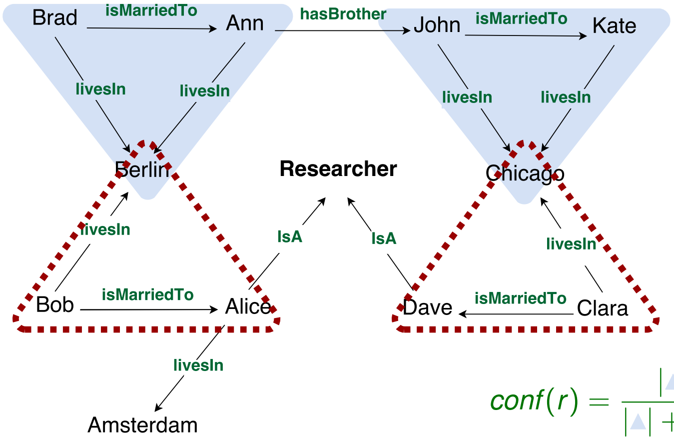
CWA: Whatever is not known is false.



Horn Rule Learning from KGs

WARMER: confidence

CWA: Whatever is not known is false.



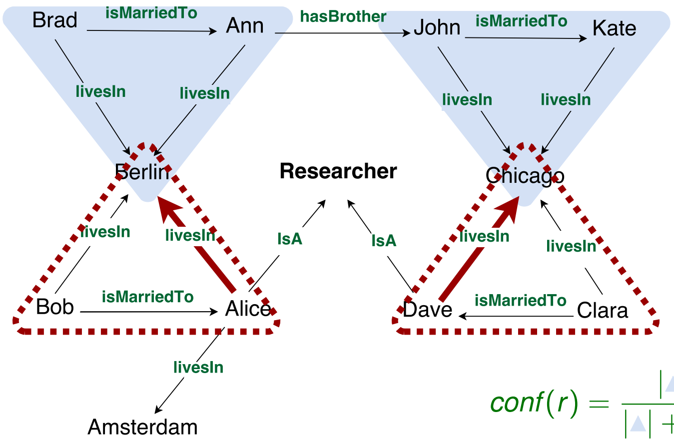
$$conf(r) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

$$r: \ livesIn(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z)$$

Horn Rule Learning from KGs

WARMER: confidence

CWA: Whatever is not known is false.



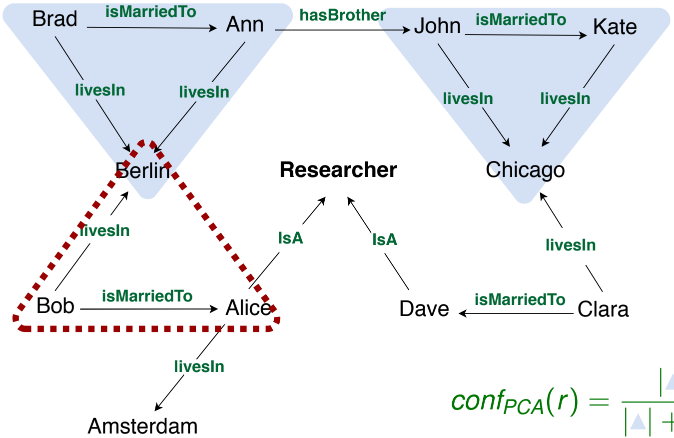
$$conf(r) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

$$r : \; livesIn(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z)$$

Horn Rule Learning from KGs

AMIE [Galarraga *et al.*, 2015]: PCA confidence

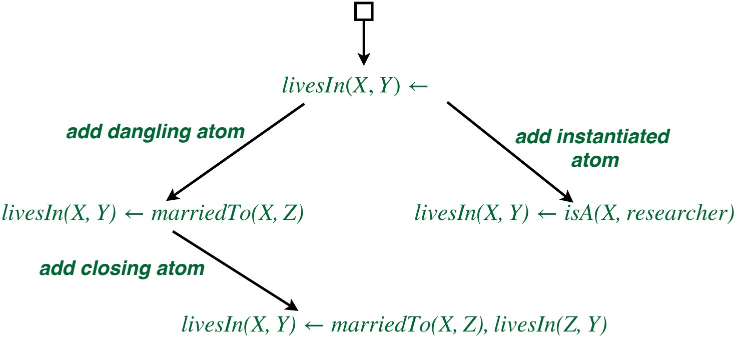
PCA: If at least 1 living place of Alice is known, then all are known.



$$conf_{PCA}(r) = \frac{|\triangle|}{|\triangle| + |\triangle|} = \frac{2}{3}$$

$$r: \ livesIn(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z)$$

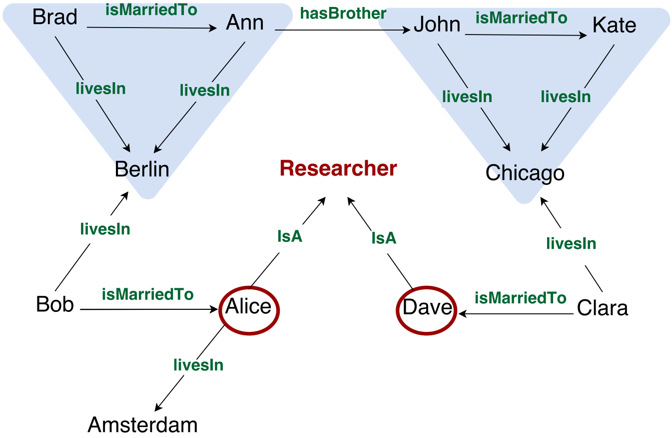
AMIE Refinement Operators



https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/amie/

Nonmonotonic Rule Learning

Nonmonotonic rule mining from KGs: OWA is a challenge!



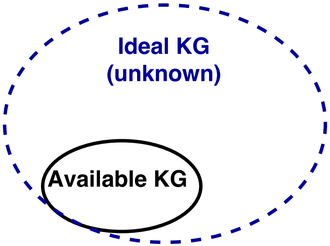
$$r: \ livesIn(X,Z) \leftarrow isMarriedTo(Y,X), livesIn(Y,Z), not \ \ researcher(X)$$

Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG

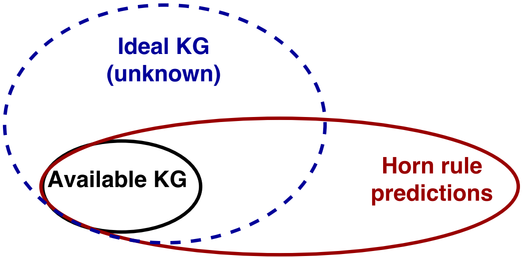


Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set

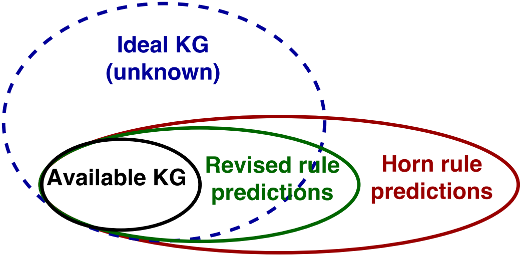


Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set



Find:

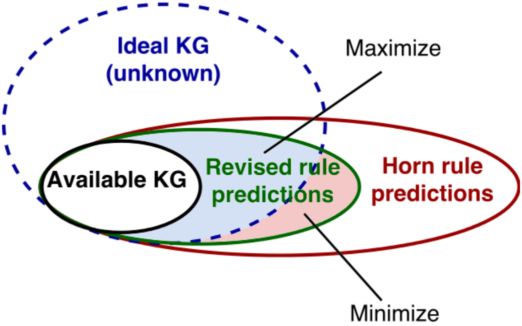
- Nonmonotonic revision of Horn rule set

Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set



Find:

- Nonmonotonic revision of Horn rule set
  with better predictive quality

Avoid Data Overfitting

How to distinguish exceptions from noise?

$$r1: \ livesIn(X,Z) \leftarrow isMarriedTo(Y,X), livesIn(Y,Z), not \ \ researcher(X)$$

Avoid Data Overfitting

How to distinguish exceptions from noise?

$r1: \ livesIn(X,Z) \leftarrow isMarriedTo(Y,X), livesIn(Y,Z), not \ \ researcher(X)$

$not\_livesIn(X,Z) \leftarrow isMarriedTo(Y,X), livesIn(Y,Z), researcher(X)$

Avoid Data Overfitting

How to distinguish exceptions from noise?

$$r1: livesIn(X,Z) \leftarrow isMarriedTo(Y,X), livesIn(Y,Z), not\ \ researcher(X)$$
$$not\_livesIn(X,Z) \leftarrow isMarriedTo(Y,X), livesIn(Y,Z), researcher(X)$$

$$r2: livesIn(X,Z) \leftarrow bornIn(X,Z), not\ \ moved(X)$$
$$not\_livesIn(X,Z) \leftarrow bornIn(X,Z), moved(X)$$

Avoid Data Overfitting

How to distinguish exceptions from noise?

$r1: \; livesIn(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z), not \;\; researcher(X)$
$\quad not\_livesIn(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z), researcher(X)$

$r2: \; livesIn(X, Z) \leftarrow bornIn(X, Z), not \;\; moved(X)$
$\quad not\_livesIn(X, Z) \leftarrow bornIn(X, Z), moved(X)$

$\{livesIn(c, d), not\_livesIn(c, d)\}$ are conflicting predictions

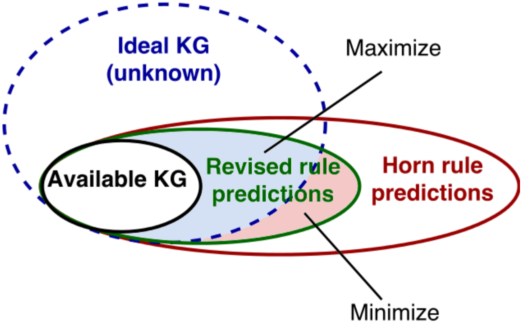
Intuition: Rules with good exceptions should make few conflicting predictions

Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set

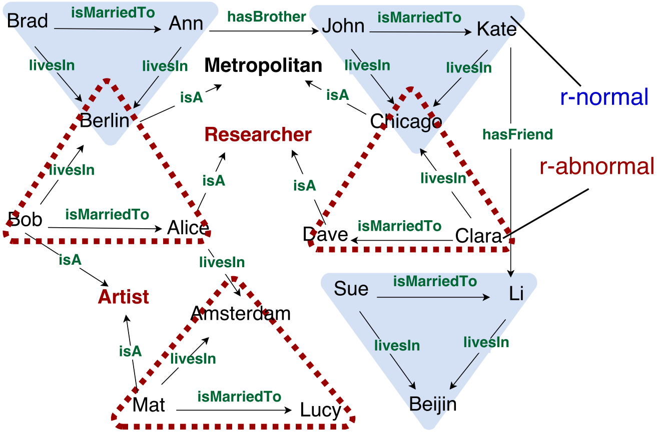


Find:

- Nonmonotonic revision of Horn rules, such that
  - number of conflicting predictions is **minimal**
  - average conviction is **maximal**

M. Gad-Elrab, D. Stepanova, J. Urbani, G. Weikum. Exception-enriched Rule Learning from Knowledge Graphs. ISWC2016
D. Tran, D. Stepanova, M. Gad-Elrab, F. Lisi, G. Weikum. Towards Nonmonotonic Relational Learning from KGs. ILP2016

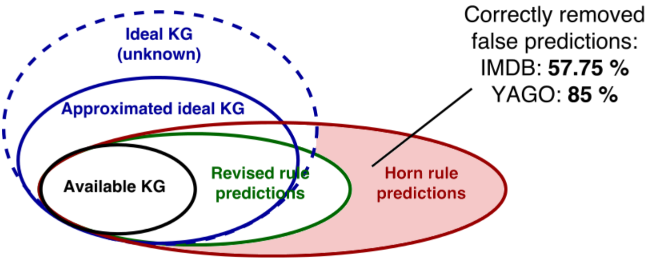
Exception Candidates



$$r\colon livesIn(X,Z) \leftarrow isMarriedTo(Y,X),\ livesIn(Y,Z) \qquad \left\{\begin{array}{l} not\ \ researcher(X) \\ not\ \ artist(Y) \end{array}\right\}$$

Experiments

- Approximated ideal KG: original KG
- Available KG: for every relation randomly remove 20% of facts from approximated ideal KG
- Horn rules: $h(X, Y) \leftarrow p(X, Z), q(Z, Y)$
- Exceptions: $e_1(X), e_2(Y), e_3(X, Y)$
- Predictions are computed using answer set solver DLV



https://github.com/htran010589/nonmonotonic-rule-mining.git

Experiments

- Approximated ideal KG: original KG
- Available KG: for every relation randomly remove 20% of facts from approximated ideal KG
- Horn rules: $h(X, Y) \leftarrow p(X, Z), q(Z, Y)$
- Exceptions: $e_1(X), e_2(Y), e_3(X, Y)$
- Predictions are computed using answer set solver DLV

Examples of revised rules:

Plots of films in a sequel are written by the same writer, unless a film is American

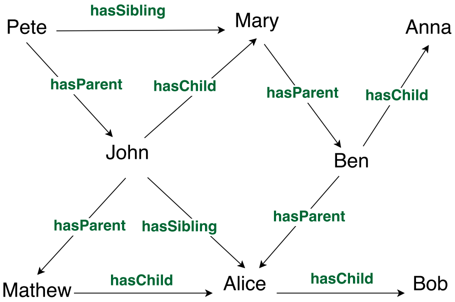
$r_1 : \mathit{writtenBy}(X, Z) \leftarrow \mathit{hasPredecessor}(X, Y), \mathit{writtenBy}(Y, Z), \mathit{not}\ \mathit{american\_film}(X)$

Spouses of film directors appear on the cast, unless they are silent film actors

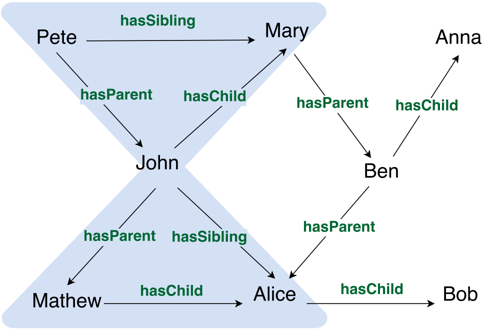
$r_2 : \mathit{actedIn}(X, Z) \leftarrow \mathit{isMarriedTo}(X, Y), \mathit{directed}(Y, Z), \mathit{not}\ \mathit{silent\_film\_actor}(X)$

https://github.com/htran010589/nonmonotonic-rule-mining.git

Reasonable Rules

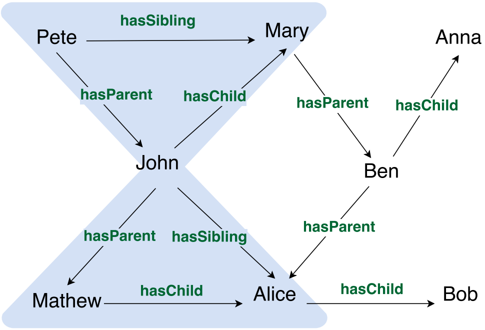


Reasonable Rules



Reasonable Rules

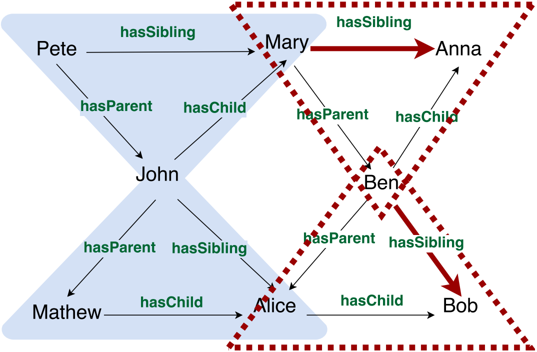
✓ *People with the same parents are likely siblings*



$$r_1: \; hasSibling(X, Z) \leftarrow hasParent(X, Y), hasChild(Y, Z)$$

Reasonable Rules

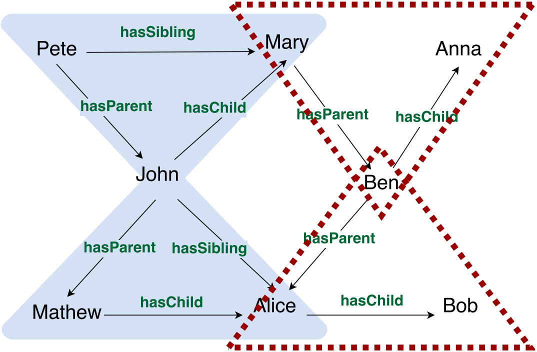
✓ *People with the same parents are likely siblings*



$$r_1 : \; hasSibling(X, Z) \leftarrow hasParent(X, Y), hasChild(Y, Z)$$

Reasonable Rules

✓ *People with the same parents are likely siblings*

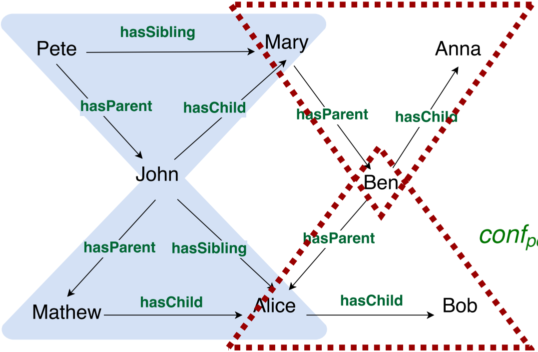


$$\mathit{conf}(r_1) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

$$r_1 : \ \mathit{hasSibling}(X, Z) \leftarrow \mathit{hasParent}(X, Y), \mathit{hasChild}(Y, Z)$$

Reasonable Rules

✓ *People with the same parents are likely siblings*

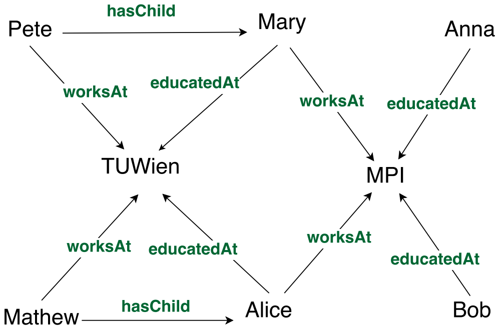


$$conf(r_1) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

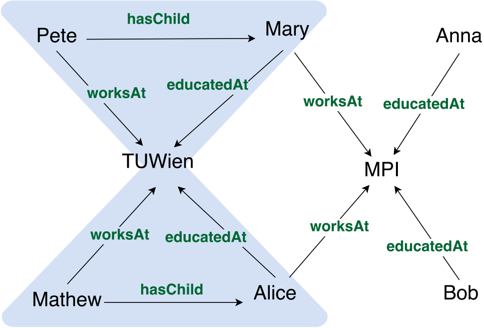
$$conf_{pca}(r_1) = \frac{|\blacktriangle|}{|\{\triangle | hasSibling(X, \_) \in \mathcal{G}\}|} = \frac{2}{2}$$

$$r_1 : \; hasSibling(X, Z) \leftarrow hasParent(X, Y), hasChild(Y, Z)$$

Erroneous Rules due to Data Bias

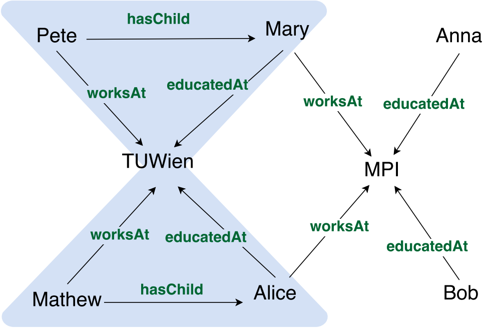


Erroneous Rules due to Data Bias



Erroneous Rules due to Data Bias

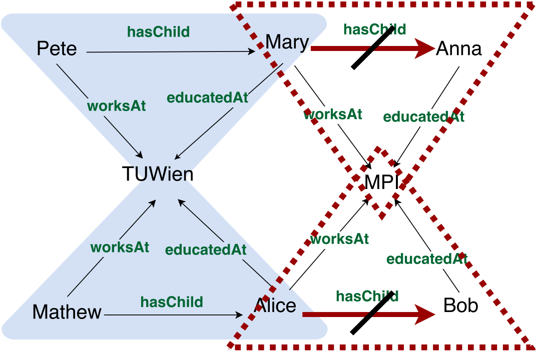
✗ *If one is studying in a university where you teach, he/she is your child*



$$r_2: \ hasChild(X, Z) \leftarrow worksAt(X, Y), educatedAt(Z, Y)$$

Erroneous Rules due to Data Bias

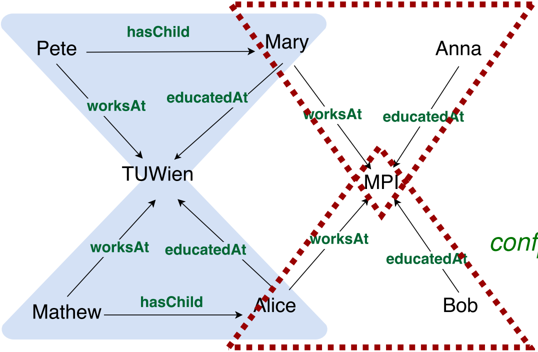
✗ *If one is studying in a university where you teach, he/she is your child*



r_2 : $hasChild(X, Z) \leftarrow worksAt(X, Y), educatedAt(Z, Y)$

Erroneous Rules due to Data Bias

✗ *If one is studying in a university where you teach, he/she is your child*



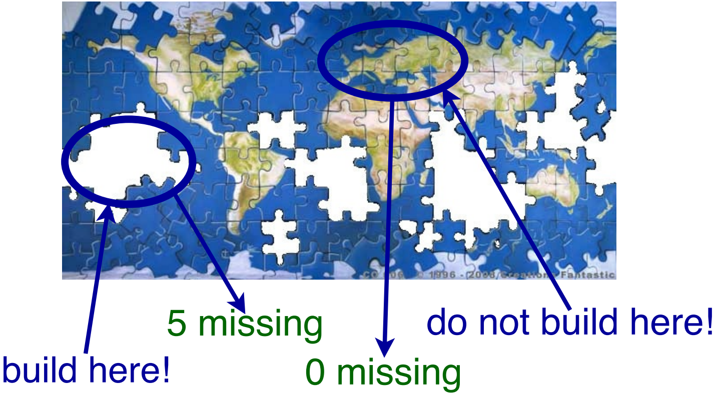
$$conf(r_2) = \frac{|\blacktriangle|}{|\blacktriangle| + |\text{♻}|} = \frac{2}{4}$$

$$conf_{pca}(r_2) = \frac{|\blacktriangle|}{|\{\text{♻} \mid hasChild(X, \_) \in \mathcal{G}\}|} = \frac{2}{2}$$

$$r_2: \; hasChild(X, Z) \leftarrow worksAt(X, Y), educatedAt(Z, Y)$$

Exploiting Meta-data in Rule Learning

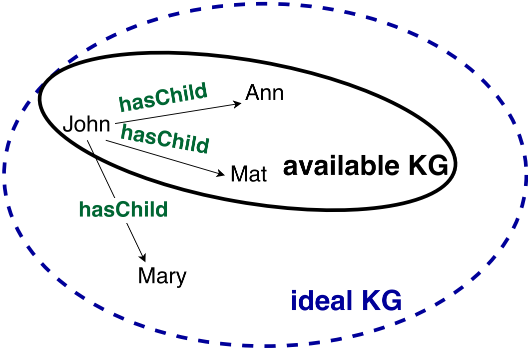
Goal: make use of cardinality constraints on edges of the KG to improve rule learning.



T. Pellissier-Tanon, D. Stepanova, S. Razniewski, P. Mirza, G. Weikum. Completeness-aware rule learning from KGs. ISWC2017.

Cardinality Statements

- $num(p, s)$: Number of outgoing p-edges from s in the ideal KG
- $miss(p, s)$: Number of missing p-edges from s in the available KG
- If $miss(p, s) = 0$, then $complete(p, s)$, otherwise $incomplete(p, s)$



$num(hasChild, john) = 3$
$miss(hasChild, john) = 1$
$incomplete(hasChild, john)$

Cardinality Constraints on Edges

- Mining cardinality assertions from the Web [Mirza *et al.*, 2016]
  - *"... John has **2** children ..."*
- Estimating recall of KGs by crowd sourcing [Razniewski *et al.*, 2016]
  - *20 % of Nobel laureates in physics are missing*
- Predicting completeness in KGs [Galárraga *et al.*, 2017]
  - Add $complete(john, hasChild)$ to KG and mine rules
    $complete(X, hasChild) \leftarrow child(X)$

Completeness Confidence

$conf_{comp}$: do not penalize rules that predict new facts in incomplete areas

$$conf_{comp}(r) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle| - npi(r)}$$

- $npi(r)$: number of facts added to incomplete areas by r
- Generalizes standard confidence ($miss(r) = 0$)
- Generalizes PCA confidence ($miss(r) \in \{0, +\infty\}$)

Other Completeness-aware Measures

$precision_{comp}$: penalize r that predict facts in complete areas

$$precision_{comp}(r) = 1 - \frac{npc(r)}{|\blacktriangle| + |\text{♻}|}$$

$recall_{comp}$: ratio of missing facts filled by r

$$recall_{comp}(r) = \frac{npi(r)}{\sum_s miss(h, s)}$$

dir_metric : proportion of predictions in complete and incomplete parts

$$dir\_metric(r) = \frac{npi(r) - npc(r)}{2 \cdot (npi(r) + npc(r))} + 0.5$$

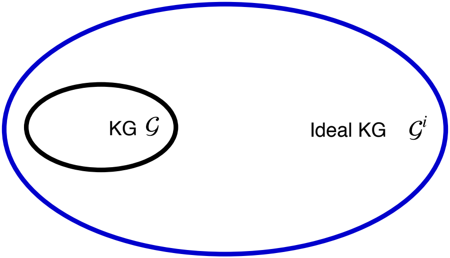
wdm : weighted combination of confidence and directional metric

$$wdm(r) = \beta \cdot conf(r) + (1 - \beta) \cdot dir\_metric(r)$$

`https://github.com/Tpt/CARL.git`

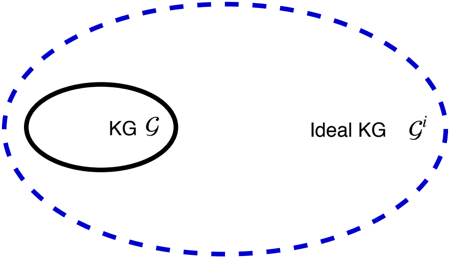
Ideal KG

$\mu(r, \mathcal{G}^i)$: measure quality of the rule r on $\mathcal{G}^i$



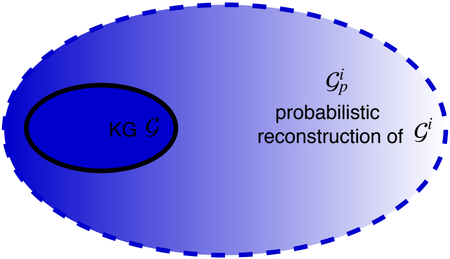
Ideal KG

$\mu(r, \mathcal{G}^i)$: measure quality of the rule r on $\mathcal{G}^i$, but $\mathcal{G}^i$ is unknown



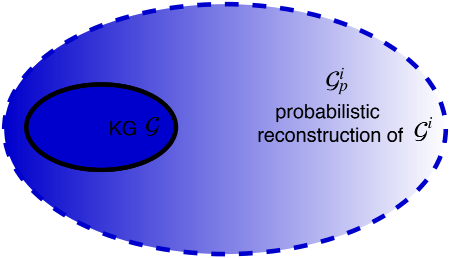
Probabilistic Reconstruction of Ideal KG

$\mu(r, \mathcal{G}_p^i)$: measure quality of r on $\mathcal{G}_p^i$



Hybrid Rule Measure

$$\mu(r, \mathcal{G}_p^i) = (1 - \lambda) \times \mu_1(r, \mathcal{G}) + \lambda \times \mu_2(r, \mathcal{G}_p^i)$$



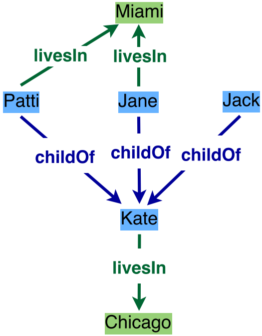
Hybrid Rule Measure

$$\mu(r, \mathcal{G}_p^i) = (1-\lambda) \times \mu_1(r, \mathcal{G}) + \lambda \times \mu_2(r, \mathcal{G}_p^i)$$

- **$\lambda \in [0..1]$: weighting factor**
- **μ_1 : descriptive quality** of rule r over the available KG $\mathcal{G}$
  - confidence
  - PCA confidence
- **μ_2 : predictive quality** of r relying on $\mathcal{G}_p^i$ (probabilistic reconstruction of the ideal KG $\mathcal{G}^i$)

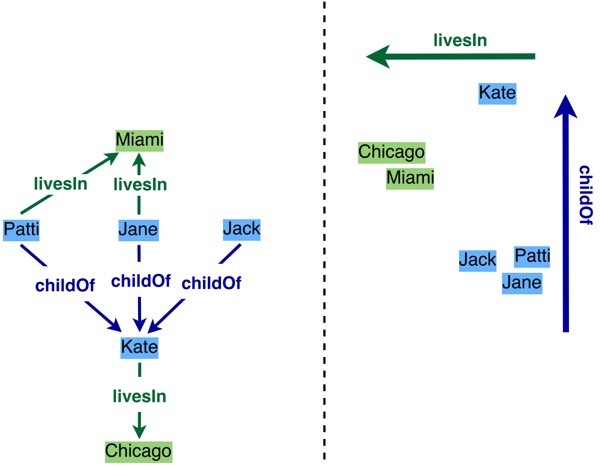
KG Embeddings

- **Intuition:** For $\langle s, p, o \rangle$ in KG, find $\mathbf{s}, \mathbf{p}, \mathbf{o}$ such that $\mathbf{s} + \mathbf{p} \approx \mathbf{o}$
- The "error of translation" of a true KG fact should be smaller by a certain margin than the "error of translation" of an out-of-KG one



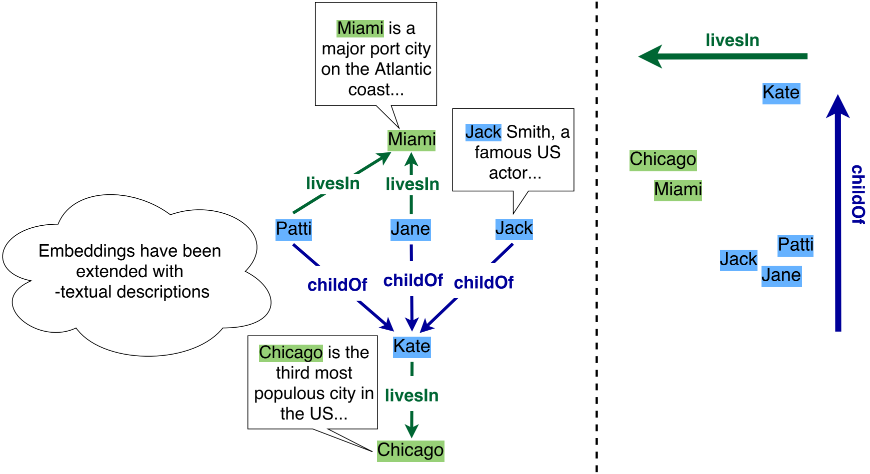
KG Embeddings

- **Intuition:** For $\langle s, p, o\rangle$ in KG, find $\mathbf{s}, \mathbf{p}, \mathbf{o}$ such that $\mathbf{s} + \mathbf{p} \approx \mathbf{o}$
- The "error of translation" of a true KG fact should be smaller by a certain margin than the "error of translation" of an out-of-KG one



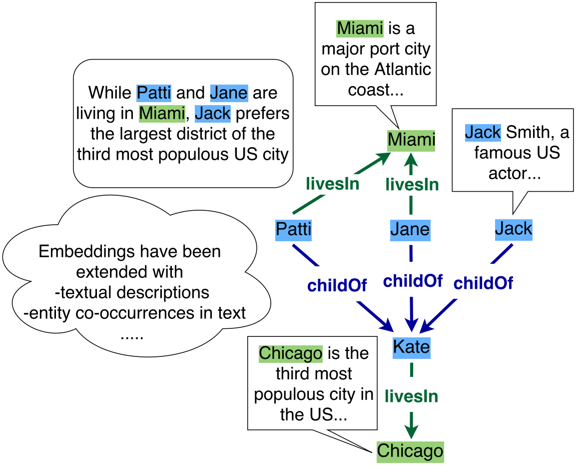
KG Embeddings

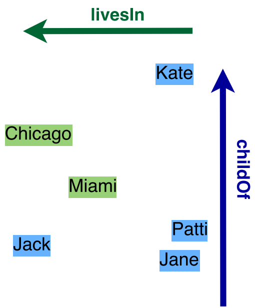
- **Intuition:** For $\langle s, p, o \rangle$ in KG, find $\mathbf{s}, \mathbf{p}, \mathbf{o}$ such that $\mathbf{s} + \mathbf{p} \approx \mathbf{o}$
- The "error of translation" of a true KG fact should be smaller by a certain margin than the "error of translation" of an out-of-KG one



KG Embeddings

- **Intuition:** For $\langle s, p, o \rangle$ in KG, find $\mathbf{s}, \mathbf{p}, \mathbf{o}$ such that $\mathbf{s} + \mathbf{p} \approx \mathbf{o}$
- The "error of translation" of a true KG fact should be smaller by a certain margin than the "error of translation" of an out-of-KG one





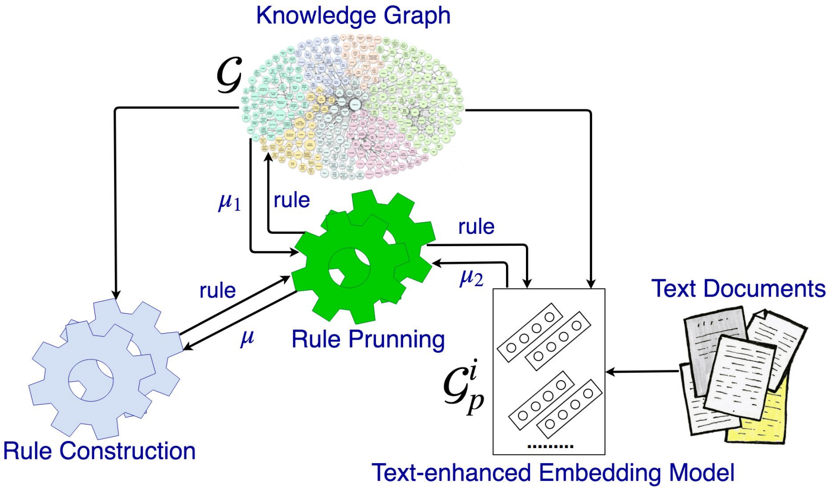
rank(<Jack livesIn Chicago>) = 0.8

rank(<Jack livesIn Miami>) = 0.6

...

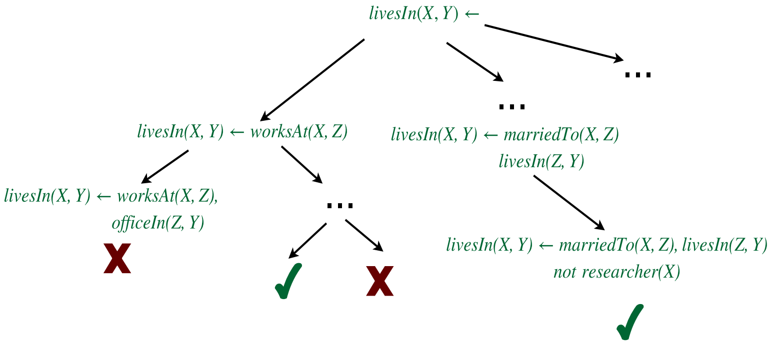
rank(<Jack livesIn NewYork>) = 0.2

Embedding-based Rule Learning



V. T. Ho, D. Stepanova, M. Gad-Elrab, E. Kharlamov, G. Weikum. Rule Learning from KGs Guided by Embedding Models. ISWC 2018

Rule Prunning



Prune rule search space relying on

- novel hybrid embedding-based rule measure

Evaluation Setup

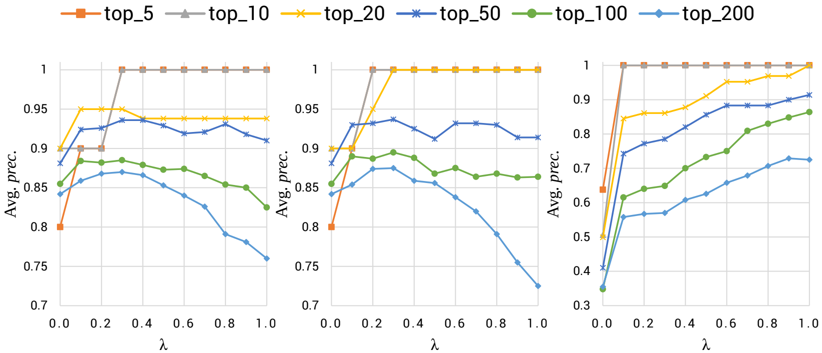
- Datasets:
  - FB15K: 592K facts, 15K entities and 1345 relations
  - Wiki44K: 250K facts, 44K entities and 100 relations
- Training graph $\mathcal{G}$: remove 20% from the available KG
- Embedding models $\mathcal{G}_p^i$:
  - TransE [Bordes *et al.*, 2013], HolE [Nickel *et al.*, 2016]
  - With text: SSP [Xiao *et al.*, 2017]
- Goals:
  - Evaluate effectiveness of our hybrid rule measure

$$\mu(r, \mathcal{G}_p^i) = (1 - \lambda) \times \mu_1(r, \mathcal{G}) + \lambda \times \mu_2(r, \mathcal{G}_p^i)$$

  - Compare against state-of-the-art rule learning systems

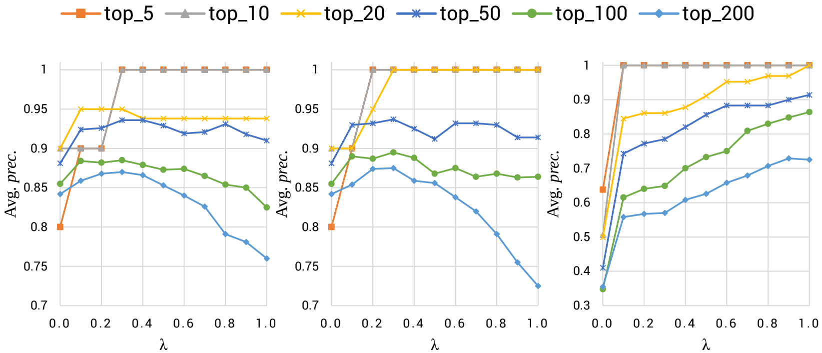
https://github.com/hovinhthinh/RuLES.git

Evaluation of Hybrid Rule Measure



(a) Conf-HolE (b) Conf-SSP (c) PCA-SSP

Evaluation of Hybrid Rule Measure



(a) Conf-HolE (b) Conf-SSP (c) PCA-SSP

- Positive impact of embeddings in all cases for $\lambda = 0.3$
- **Note:** in (c) comparison to AMIE [Galarraga *et al.*, 2015] ($\lambda = 0$)

Example Rules

Examples of rules learned from Wikidata

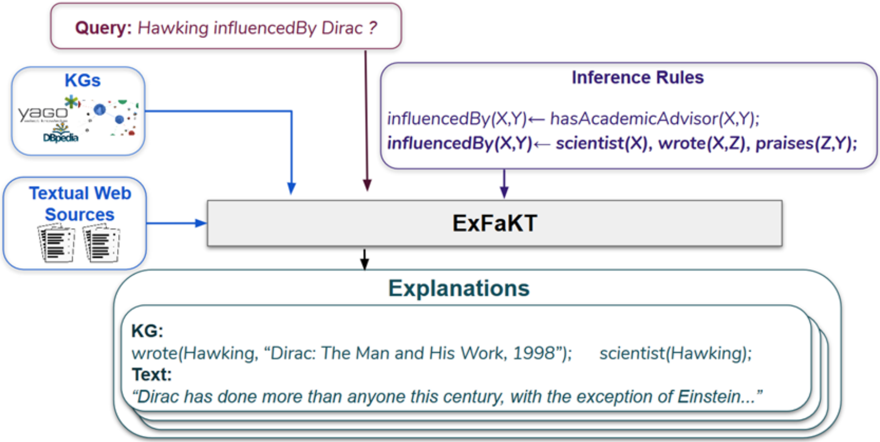
Script writers stay the same throughout a sequel, but not for TV series

r_1 : $\mathit{scriptwriterOf}(X, Y) \leftarrow \mathit{precededBy}(Y, Z), \mathit{scriptwriterOf}(X, Z),$ **not** $\mathit{isA}(Z, \mathit{tvSeries})$

Nobles are typically married to nobles, but not in the case of Chinese dynasties

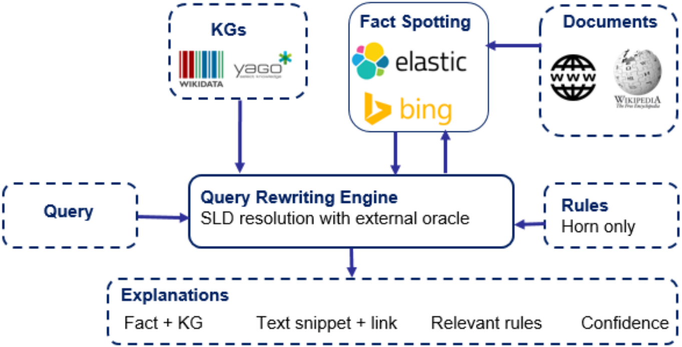
r_2 : $\mathit{nobleFamily}(X, Y) \leftarrow \mathit{spouse}(X, Z), \mathit{nobleFamily}(Z, Y),$ **not** $\mathit{isA}(Y, \mathit{chineseDynasty})$

Rule-based Fact Checking



M. Gad-Elrab, D. Stepanova, J. Urbani, G. Weikum. ExFakt: A Framework for Explaining Facts over KGs and Text. WSDM 2019.
M. Gad-Elrab, D. Stepanova, J. Urbani, G. Weikum. Tracy: Tracing Facts over Knowledge Graphs and Text. WWW 2019.

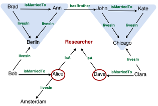
Rule-based Fact Checking



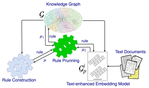
M. Gad-Elrab, D. Stepanova, J. Urbani, G. Weikum. ExFakt: A Framework for Explaining Facts over KGs and Text. WSDM 2019.
M. Gad-Elrab, D. Stepanova, J. Urbani, G. Weikum. Tracy: Tracing Facts over Knowledge Graphs and Text. WWW 2019.

Summary

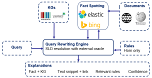
- Classical rule learning methods from ILP



- Rule learning from Knowledge Graphs
- Exploiting embeddings to guide rule learning

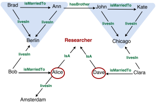


- Rule-based fact checking

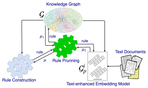


Summary

- Classical rule learning methods from ILP

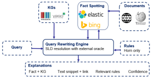


- Rule learning from Knowledge Graphs



- Exploiting embeddings to guide rule learning

- Rule-based fact checking



Interested in PhD/internship?
At BCAI we are hiring!

jobs.smartrecruiters.com/BoschGroup/743999698936956-phd-combined-reasoning-and-learning-approaches

Huge Thanks!

- For collaborations on the presented work:
  - Mohamed Gad-elrab, Thinh Vinh Ho, Hai Dang Tran, Thomas Pellissier-Tanon, Gerhard Weikum, Jacopo Urbani, Evgeny Kharlamov, Francesca A. Lisi, Simon Razniewski, Paramita Mirza
- For fruitful discussions and/or making slides available online:
  - Thomas Eiter, Stephen Muggleton, Luc De Raedt, Fabian Suchanek

References I

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko.
Translating Embeddings for Modeling Multi-relational Data.
In *Proceedings of NIPS*, pages 2787–2795, 2013.

Claudia d'Amato, Steffen Staab, Andrea GB Tettamanzi, Tran Duc Minh, and Fabien Gandon.
Ontology enrichment by discovering multi-relational association rules from ontological knowledge bases.
In *SAC*, pages 333–338, 2016.

Richard Evans and Edward Grefenstette.
Learning explanatory rules from noisy data.
J. Artif. Intell. Res., 61:1–64, 2018.

Luis Galarraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek.
Fast rule mining in ontological knowledge bases with AMIE+.
In *VLDB*, volume 24, pages 707–730, 2015.

Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M Suchanek.
Predicting completeness in knowledge bases.
WSDM, 2017.

Bart Goethals and Jan Van den Bussche.
Relational association rules: Getting warmer.
In *PDD*, 2002.

Nikos Katzouris, Alexander Artikis, and Georgios Paliouras.
Incremental learning of event definitions with inductive logic programming.
Machine Learning, 100(2-3):555–585, 2015.

Mark Law, Alessandra Russo, and Krysia Broda.
The ILASP system for learning answer set programs.
`https://www.doc.ic.ac.uk/~ml1909/ILASP`, 2015.

References II

Jens Lehmann.
DL-Learner: Learning concepts in description logics.
Journal of Machine Learning Research, pages 2639–2642, 2009.

Francesca A. Lisi.
Inductive Logic Programming in Databases: From Datalog to DL+log.
TPLP, 10(3):331–359, 2010.

Paramita Mirza, Simon Razniewski, and Werner Nutt.
Expanding wikidata's parenthood information by 178%, or how to mine relation cardinality information.
In *ISWC 2016 Posters & Demos*, 2016.

Stephen Muggleton.
Inductive logic programming.
New Generation Comput., 8(4):295–318, 1991.

Stephen Muggleton.
Inverse entailment and progol.
New Generation Comput., 13(3&4):245–286, 1995.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio.
Holographic embeddings of knowledge graphs.
In *AAAI*, 2016.

Luc De Raedt and Saso Dzeroski.
First-order jk-clausal theories are pac-learnable.
Artif. Intell., 70(1-2):375–392, 1994.

Simon Razniewski, Fabian M. Suchanek, and Werner Nutt.
But what do we actually know?
In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction, AKBC@NAACL-HLT 2016, San Diego, CA, USA, June 17, 2016*, pages 40–44, 2016.

References III

Chiaki Sakama.
Induction from answer sets in nonmonotonic logic programs.
ACM Trans. Comput. Log., 6(2):203–231, 2005.

Ehud Y. Shapiro.
Inductive inference of theories from facts.
In *Computational Logic - Essays in Honor of Alan Robinson*, pages 199–254, 1991.

Han Xiao, Minlie Huang, Lian Meng, and Xiaoyan Zhu.
SSP: semantic space projection for knowledge graph embedding with text descriptions.
In *AAAI*, 2017.