Rule Induction and Reasoning over Knowledge Graphs

Daria Stepanova, Mohamed Gad-elrab, Vinh Thinh Ho

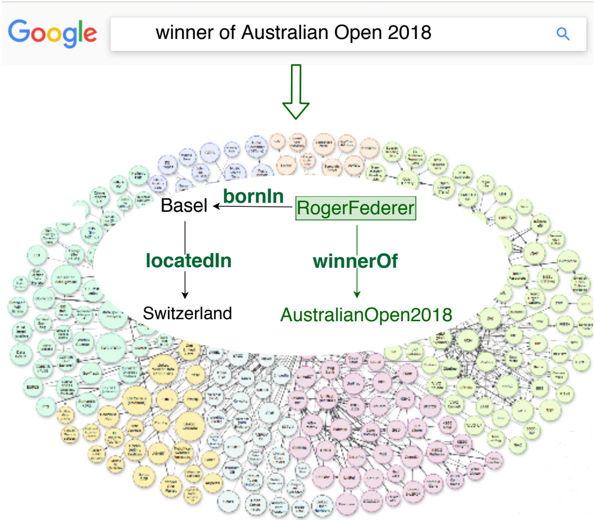
Max Planck Institute for Informatics, Saarbrücken, Germany

25.09.2018

Outline

Semantic Web Search

Semantic Web Search



Semantic Web Search

living place of Roger Federer

All Images News Videos Shopping More Settings Tools

About 2.690.000 results (0,55 seconds)

Roger Federer's glass mansion: Tennis star's £6.5m Swiss waterfront ...
www.telegraph.co.uk › Sport › Tennis › Roger Federer ▾
Tennis star **Roger Federer** is to move his family into a £6.5million glass mansion on the shores of Lake Zurich after work was completed on the state-of-the-art ...

Roger Federer's Luxurious Houses | Basel Shows
www.baselshows.com/basel-world/the-houses-of-roger-federer ▾
Roger Federer also owns a lavish apartment in Dubai apart from properties in Switzerland. He has chosen this **location** as a base of training to get use to heat ...

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

$$livesIn(Y,Z) \leftarrow marriedTo(X,Y), livesIn(X,Z)$$

Married people live together

$$marriedTo(mirka, roger)$$

Mirka is married to Roger

$$livesIn(mirka, bottmingen)$$

Mirka lives in Bottmingen

$$livesIn(roger, bottmingen)$$

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

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

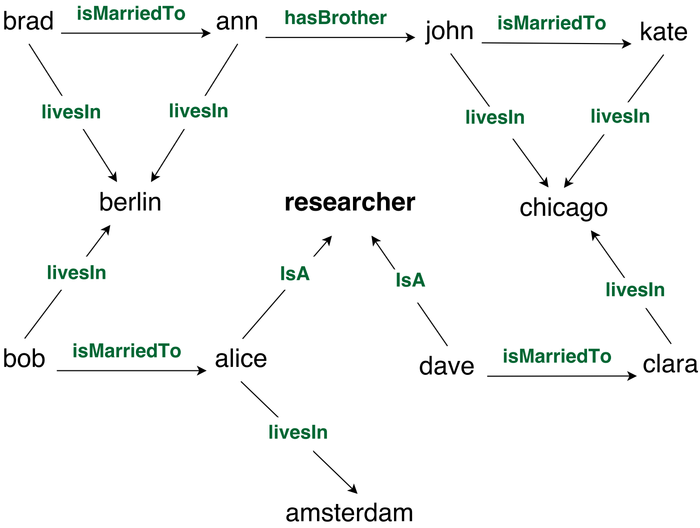
Mirka lives in Bottmingen

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

Roger lives in Bottmingen

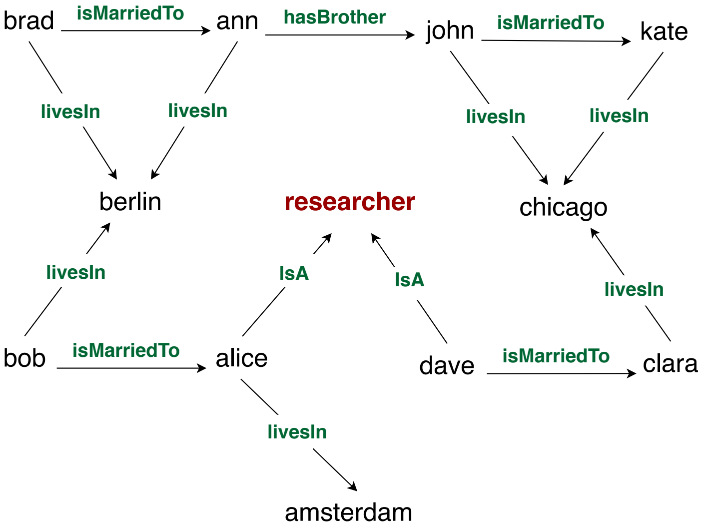
But where can a machine get such rules from?

Inducing Rules from KGs



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

Inducing Rules from KGs



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

Applications of Rule Learning

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

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

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

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

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
$walk \leftarrow at(L), crossing(L), \neg train\_approaches(L)$

Herbrand Semantics

Herbrand universe of a logic program P, $HU(P)$ is the set of all constants appearing in P.

Herbrand base of P, $HB(P)$ is the set of all ground atoms which can be formed from predicates and constants of P.

(Herbrand) interpretation of P, I is a subset of the Herbrand base.

Example: Herbrand universe, base, interpretation

$$P = \begin{cases} (1)\ isMarriedTo(mirka, roger) \quad (2)\ livesIn(mirka, bottmingen) \\ (3)\ livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), livesIn(X, Z), not\ \ researcher(Y) \end{cases}$$

Herbrand Semantics

Herbrand universe of a logic program P, $HU(P)$ is the set of all constants appearing in P.

Herbrand base of P, $HB(P)$ is the set of all ground atoms which can be formed from predicates and constants of P.

(Herbrand) interpretation of P, I is a subset of the Herbrand base.

Example: Herbrand universe, base, interpretation

$$P = \begin{cases} (1)\ isMarriedTo(mirka, roger) \quad (2)\ livesIn(mirka, bottmingen) \\ (3)\ livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), livesIn(X, Z), not\ \ researcher(Y) \end{cases}$$

$HU(P) = \{mirka, roger, bottmingen\}$

Herbrand Semantics

Herbrand universe of a logic program P, $HU(P)$ is the set of all constants appearing in P.

Herbrand base of P, $HB(P)$ is the set of all ground atoms which can be formed from predicates and constants of P.

(Herbrand) interpretation of P, I is a subset of the Herbrand base.

Example: Herbrand universe, base, interpretation

$$P = \begin{cases} (1)\ isMarriedTo(mirka, roger) \quad (2)\ livesIn(mirka, bottmingen) \\ (3)\ livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), livesIn(X, Z), not\ \ researcher(Y) \end{cases}$$

$$HU(P) = \{mirka, roger, bottmingen\}$$

$$\begin{aligned} HB(P) = \{&isMarriedTo(mirka, mirka), isMarriedTo(mirka, roger), \ldots \\ &livesIn(mirka, bottmingen), livesIn(roger, bottmingen), \ldots\} \end{aligned}$$

Herbrand Semantics

Herbrand universe of a logic program P, $HU(P)$ is the set of all constants appearing in P.

Herbrand base of P, $HB(P)$ is the set of all ground atoms which can be formed from predicates and constants of P.

(Herbrand) interpretation of P, I is a subset of the Herbrand base.

Example: Herbrand universe, base, interpretation

$$P = \left\{ \begin{array}{l} (1)\ isMarriedTo(mirka, roger) \quad (2)\ livesIn(mirka, bottmingen) \\ (3)\ livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), livesIn(X, Z), not\ \ researcher(Y) \end{array} \right.$$

$$HU(P) = \{mirka, roger, bottmingen\}$$

$$\begin{aligned} HB(P) = \{&isMarriedTo(mirka, mirka), isMarriedTo(mirka, roger), \ldots \\ &livesIn(mirka, bottmingen), livesIn(roger, bottmingen), \ldots\} \end{aligned}$$

$$I_1 = \emptyset, I_2 = \{isMarriedTo(mirka, roger), livesIn(bottmingen, bottmingen)\}, \ldots$$

Answer Set Semantics

Def.: **Herbrand models, answer sets**

- An interpretation I is a (Herbrand) model of (or satisfies)
  - ground rule $r: a \leftarrow b_1, \ldots, b_m, not\ \ b_{m+1}, \ldots, not\ \ b_n$, if $\{b_1, \ldots, b_m\} \subseteq I$ and $\{b_{m+1}, b_n\} \cap I = \emptyset$ imply $a \in I$ (written $I \models r$).

Answer Set Semantics

Def.: Herbrand models, answer sets

- An interpretation I is a (Herbrand) model of (or satisfies)
  - ground rule $r : a \leftarrow b_1, \ldots, b_m, not\ \ b_{m+1}, \ldots, not\ \ b_n$, if $\{b_1, \ldots, b_m\} \subseteq I$ and $\{b_{m+1}, b_n\} \cap I = \emptyset$ imply $a \in I$ (written $I \models r$).
  - a nonground rule r, symbolically $I \models r$, if $I \models r$ for every $r \in grnd(C)$;

Answer Set Semantics

Def.: Herbrand models, answer sets

- An interpretation I is a (Herbrand) model of (or satisfies)
  - ground rule $r : a \leftarrow b_1, \dots, b_m, not\ \ b_{m+1}, \dots, not\ \ b_n$, if $\{b_1, \dots, b_m\} \subseteq I$ and $\{b_{m+1}, b_n\} \cap I = \emptyset$ imply $a \in I$ (written $I \models r$).
  - a nonground rule r, symbolically $I \models r$, if $I \models r$ for every $r \in grnd(C)$;
  - a program P, symbolically $I \models P$, if $I \models C$ for every clause C in P.
- Minimal model (answer set): none of its subsets is a model.

Example

Consider program P:

$$livesIn(m,b). \qquad isMarriedTo(m,r). \qquad bornIn(m,b).$$
$$livesIn(Y,Z) \leftarrow livesIn(X,Y), isMarriedTo(Y,Z), not\ researcher(Y).$$
$$livesIn(X,Y) \leftarrow bornIn(X,Y).$$

Which of the following interpretations are models of P?

- $I_1 = \emptyset$
- $I_2 = HB(P)$
- $I_3 = \{livesIn(m,b), isMarriedTo(m,r), livesIn(r,b)\}$

Example

Consider program P:

$$livesIn(m,b). \quad isMarriedTo(m,r). \quad bornIn(m,b).$$
$$livesIn(Y,Z) \leftarrow livesIn(X,Y), isMarriedTo(Y,Z), not\ researcher(Y).$$
$$livesIn(X,Y) \leftarrow bornIn(X,Y).$$

Which of the following interpretations are models of P?

- $I_1 = \emptyset$ **no**
- $I_2 = HB(P)$
- $I_3 = \{livesIn(m,b), isMarriedTo(m,r), livesIn(r,b)\}$

Example

Consider program P:

$$livesIn(m, b). \quad isMarriedTo(m, r). \quad bornIn(m, b).$$
$$livesIn(Y, Z) \leftarrow livesIn(X, Y), isMarriedTo(Y, Z), not\ researcher(Y).$$
$$livesIn(X, Y) \leftarrow bornIn(X, Y).$$

Which of the following interpretations are models of P?

- $I_1 = \emptyset$ **no**
- $I_2 = HB(P)$ **yes**
- $I_3 = \{livesIn(m, b), isMarriedTo(m, r), livesIn(r, b)\}$

Example

Consider program P:

$$livesIn(m,b). \qquad isMarriedTo(m,r). \qquad bornIn(m,b).$$
$$livesIn(Y,Z) \leftarrow livesIn(X,Y), isMarriedTo(Y,Z), not\ researcher(Y).$$
$$livesIn(X,Y) \leftarrow bornIn(X,Y).$$

Which of the following interpretations are models of P?

- $I_1 = \emptyset$ **no**
- $I_2 = HB(P)$ **yes**
- $I_3 = \{livesIn(m,b), isMarriedTo(m,r), livesIn(r,b)\}$ **no**

Example

Consider program P:

$$\begin{aligned} livesIn(m,b). \quad isMarriedTo(m,r). \quad bornIn(m,b). \\ livesIn(Y,Z) \leftarrow livesIn(X,Y), isMarriedTo(Y,Z), not\ researcher(Y). \\ livesIn(X,Y) \leftarrow bornIn(X,Y). \end{aligned}$$

Which of the following interpretations are minimal models of P?

- $I_1 = HB(P)$
- $I_2 = \{livesIn(m,b), isMarriedTo(m,r), livesIn(r,b), bornIn(m,b)\}$

Example

Consider program P:

$$livesIn(m,b). \quad isMarriedTo(m,r). \quad bornIn(m,b).$$
$$livesIn(Y,Z) \leftarrow livesIn(X,Y), isMarriedTo(Y,Z), not\ researcher(Y).$$
$$livesIn(X,Y) \leftarrow bornIn(X,Y).$$

Which of the following interpretations are minimal models of P?

- $I_1 = HB(P)$ **no**
- $I_2 = \{livesIn(m,b), isMarriedTo(m,r), livesIn(r,b), bornIn(m,b)\}$

Example

Consider program P:

$$livesIn(m,b). \quad isMarriedTo(m,r). \quad bornIn(m,b).$$
$$livesIn(Y,Z) \leftarrow livesIn(X,Y), isMarriedTo(Y,Z), not\ researcher(Y).$$
$$livesIn(X,Y) \leftarrow bornIn(X,Y).$$

Which of the following interpretations are minimal models of P?

- $I_1 = HB(P)$ **no**
- $I_2 = \{livesIn(m,b), isMarriedTo(m,r), livesIn(r,b), bornIn(m,b)\}$ **yes**

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

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

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

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

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

$$researcher(john)$$

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

(4) $researcher(john)$

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

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 1990s:
  Muggleton, Quinlan, De Raedt, ...
- Statistical Relational Learning 2000s:
  Getoor, Koller, Domingos, Sato, ...

Learning from Examples

Inductive Learning from Examples [Muggleton, 1991]

Given:

- E^+ : positive examples (ground facts) over a relation p
- E^- : negative examples (ground facts) over p
- T : background theory (a set of facts and possibly rules)
- Language bias: syntactic restrictions on the definition of p

Learning from Examples

Inductive Learning from Examples [Muggleton, 1991]

Given:

- E^+ : positive examples (ground facts) over a relation p
- E^- : negative examples (ground facts) over p
- T : background theory (a set of facts and possibly rules)
- Language bias: syntactic restrictions on the definition of p

Find:

- Hyp : hypothesis defining p such that
  - Hyp "covers" all positive examples given T, i.e.,
    $\forall e \in E^+ : \ T \cup Hyp \models e$
  - Hyp does not "cover" any negative examples given T, i.e.,
    $\forall e \in E^- : \ T \cup Hyp \not\models e$

Example

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

Example

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 : interpretation, i.e., a set of facts over various relations
- T : background theory, i.e., a set of facts and possibly rules
- Language bias: syntactic restrictions on the target hypothesis

Learning from Interpretations

Inductive Learning from Interpretations [Raedt and Dzeroski, 1994]

Given:

- I : interpretation, i.e., a set of facts over various relations
- T : background theory, i.e., a set of facts and possibly rules
- Language bias: syntactic restrictions on the target hypothesis

Find:

- Hyp : hypothesis, such that I is a minimal model of $Hyp \cup T$

Example

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

Example

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

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. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, luxUni), researcher(Z)\},$

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. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, luxUni), researcher(Z)\},$
    $\theta = \{X/Z, Y/uniLux\}$

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. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, luxUni), researcher(Z)\}$,
    $\theta = \{X/Z, Y/uniLux\}$
- 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. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, luxUni), researcher(Z)\}$,
    $\theta = \{X/Z, Y/uniLux\}$
- 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. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, luxUni), researcher(Z)\},$
    $\theta = \{X/Z, Y/uniLux\}$
- 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. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, luxUni), researcher(Z)\}$,
    $\theta = \{X/Z, Y/uniLux\}$
- 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. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, luxUni), researcher(Z)\},$
    $\theta = \{X/Z, Y/uniLux\}$
- 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. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, luxUni), researcher(Z)\},$
    $\theta = \{X/Z, Y/uniLux\}$
- 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. $C\theta \subseteq D$
    $\{worksAt(X,Y)\} \succeq \{worksAt(Z, luxUni), researcher(Z)\}$,
    $\theta = \{X/Z, Y/uniLux\}$
- 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 $Hyp1 \cup T \models Hyp2$
    $livesIn(roger, bottmingen)$? $livesIn(roger, switzeland)$

Common Techniques in ILP

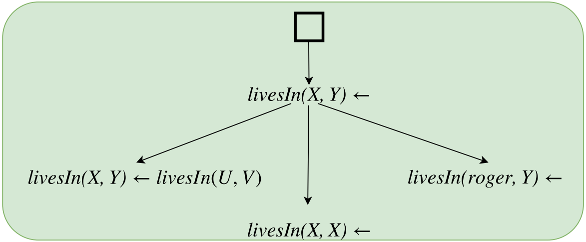
- Generality ($\succeq$): essential component of symbolic learning systems
- Genaralization as θ-subsumption
  - Atoms: $a \succeq b$ iff a substitution θ exists such that $a\theta = b$
    $person(X) \succeq person(roger), \;\; \theta = \{X/roger\}$
  - Clause: $C \succeq D$ iff θ exists, s.t. $C\theta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

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. $C\theta \subseteq D$
    $\{worksAt(X, Y)\} \succeq \{worksAt(Z, luxUni), researcher(Z)\},$
    $\theta = \{X/Z, Y/uniLux\}$
- 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 $Hyp1 \cup T \models Hyp2$
    $livesIn(roger, bottmingen) \succeq livesIn(roger, switzeland)$
    $T: \ livesIn(X, switzerland) \leftarrow livesIn(X, bottmingen)$

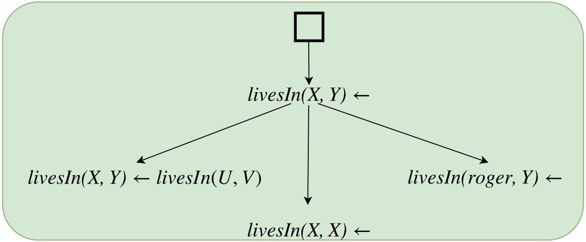
Common Techniques in ILP

- 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]

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]

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]

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

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]

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...

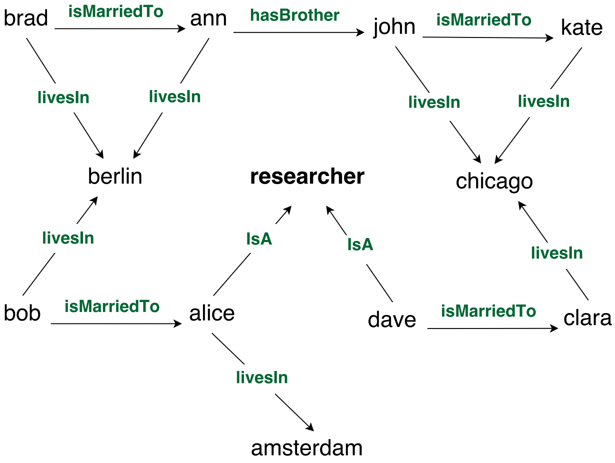
Zoo of Other ILP Tasks

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]

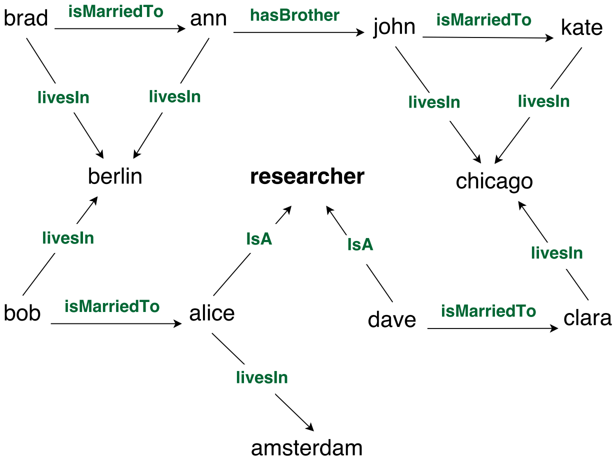
Rule Induction from Knowledge Graphs

What is the most suitable ILP task for the KG setting?



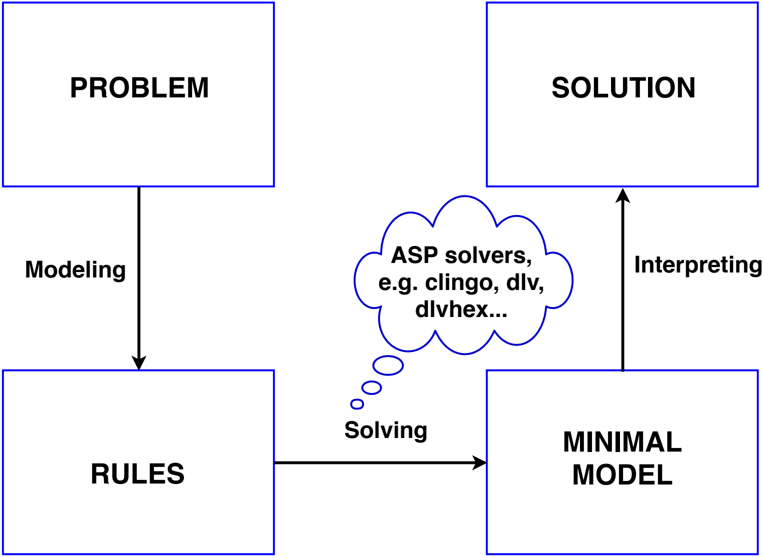
Rule Induction from Knowledge Graphs

What is the most suitable ILP task for the KG setting?



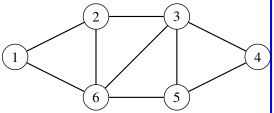
Probably learning from interpretations..

Declarative Programming



Example

Graph 3-colorability



Modeling

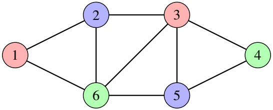
node(1 . . . 6); *edge*(1, 2); . . .
$col(V, red) \leftarrow not\ col(V, blue), not\ col(V, green), node(V);$
$col(V, green) \leftarrow not\ col(V, blue), not\ col(V, red), node(V);$
$col(V, blue) \leftarrow not\ col(V, green), not\ col(V, red), node(V);$
$\bot \leftarrow col(V, C), col(V, C'), C \neq C';$
$\bot \leftarrow col(V, C), col(V', C), edge(V, V')$

NONMONOTONIC RULES

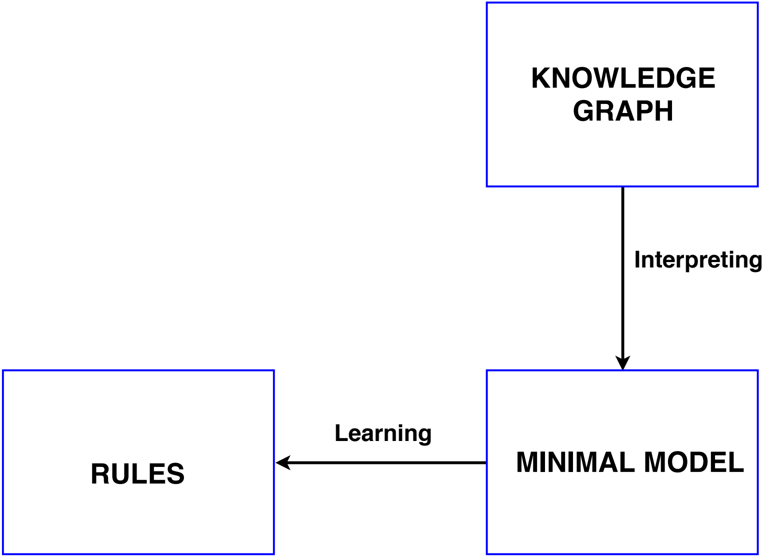
Solving

node(1 . . . 6); *edge*(1, 2); . . .
col(1, *red*), *col*(2, *blue*),
col(3, *red*), *col*(4, *green*),
col(6, *green*), *col*(5, *blue*)

Interpreting



Ideal Setting



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

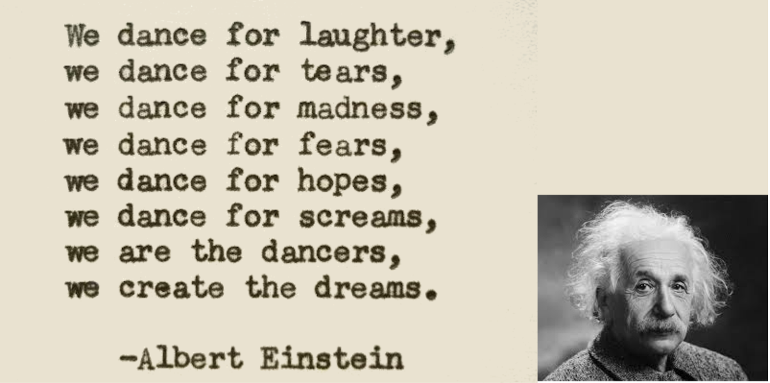
Open World Assumption: negative facts cannot be easily derived

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

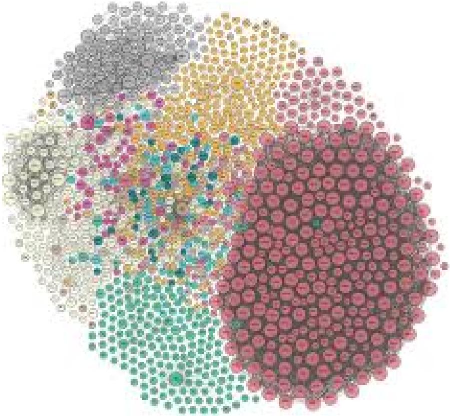
Data bias: KGs are extracted from text, which typically mentions only popular entities and interesting facts about them.

"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

Support [🍎] = 4

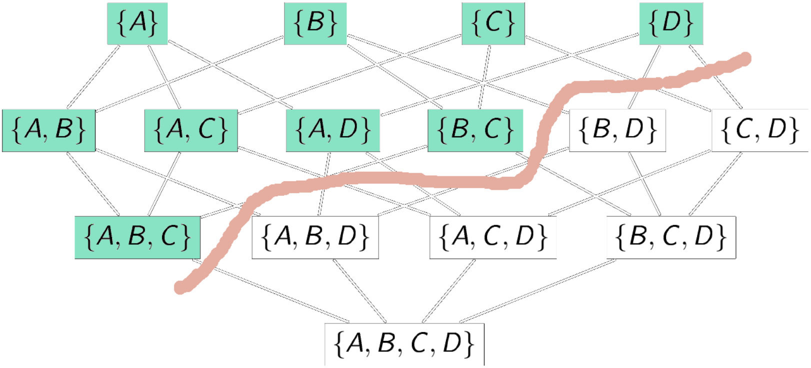
Confidence {🍎 → 🍺} = $\dfrac{\text{Support } \{🍎, 🍺\}}{\text{Support } \{🍎\}}$

Lift {🍎 → 🍺} = $\dfrac{\text{Support } \{🍎, 🍺\}}{\text{Support } \{🍎\} \times \text{Support } \{🍺\}}$

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

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



How to Apply this to Relational Data?

- DOWNGRADING DATA: Can we change the representation from richer representations to simpler ones? (So we can use systems working with simpler representations)

- UPGRADING SYSTEMS: Can we develop systems that work with richer representations (starting from systems for simpler representations)?

Downgrading the Data

- **Propositionalization** [Krogel *et al.*, 2003]: transform a KG into a transaction database

	bornInUS	*livesInUS*	*isMarriedToSinger*	*researcher*	*sportsman*
p1	✓	✓			✓
p2	✓	✓		✓	
p3	✓	✓			
p4	✓	✓			
p5	✓		✓		
p6	✓		✓		✓
p7	✓			✓	
p8	✓	✓			

Upgrading the Systems

- Start from existing system for simpler representation
- Extend it for use with richer representation (while trying to keep the original system as a special case)

Relational Association Rule Mining

- 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., $supp(q_1) = 6$

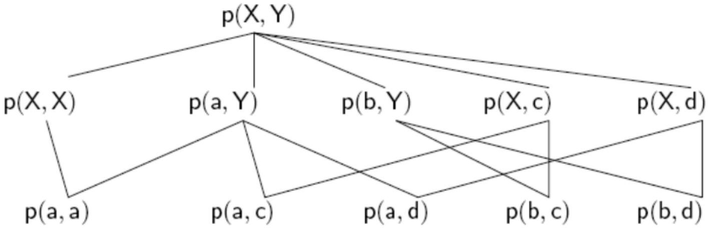
CQ: return all people with their spouses and living places

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

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

Relational Association Rule Mining

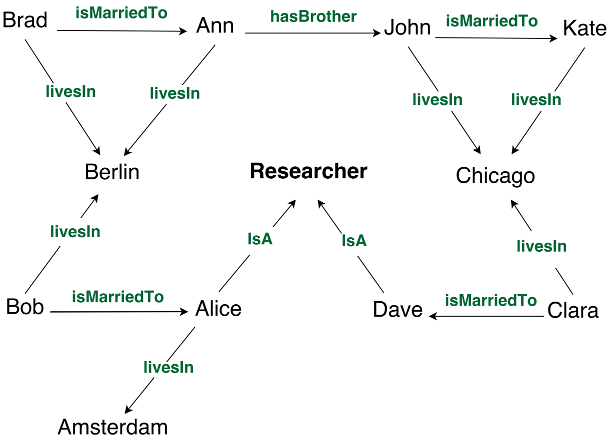
- WARMER [Goethals and den Bussche, 2002]
- Upgrade frequent itemsets to frequent conjunctive queries
    - traverse the lattice
    - get frequent CQs based on user-specified value
    - split into body and head
    - rank based on a rule measure, e.g., confidence



Horn Rule Mining from KGs

WARMER: confidence

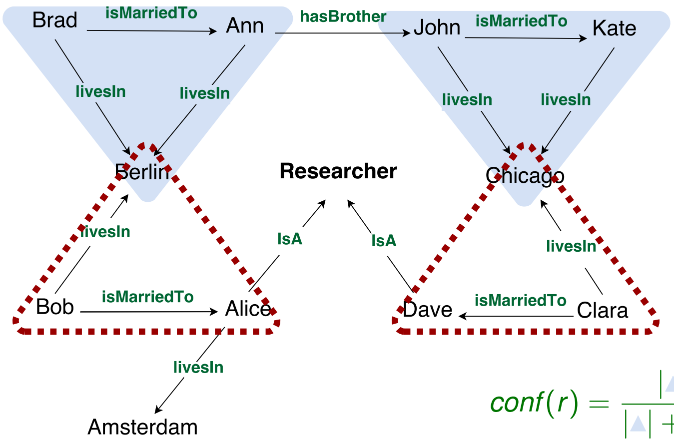
CWA: Whatever is not known is false.



Horn Rule Mining from KGs

WARMER: confidence

CWA: Whatever is not known is false.



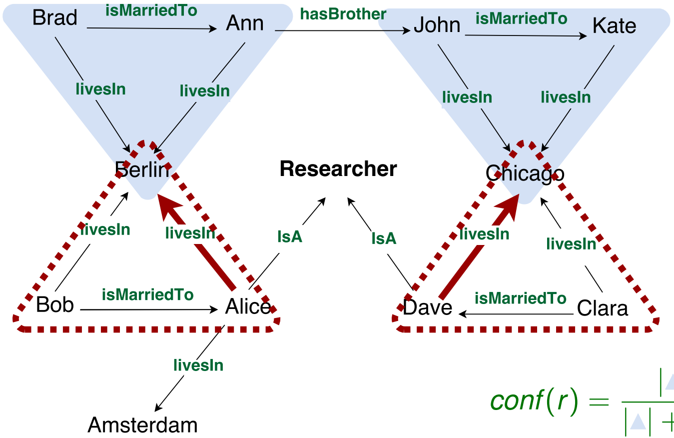
$$conf(r) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

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

Horn Rule Mining from KGs

WARMER: confidence

CWA: Whatever is not known is false.



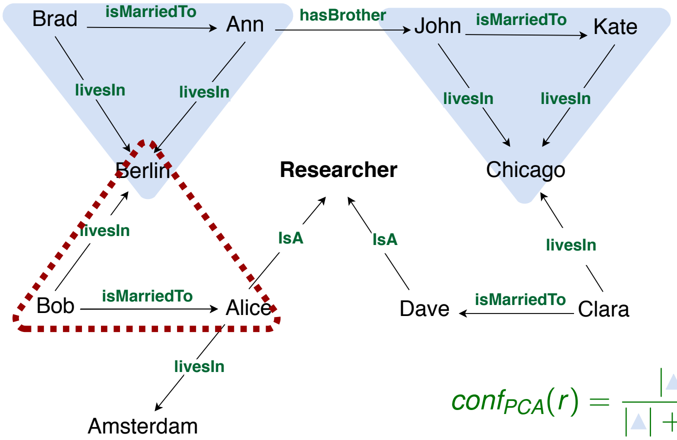
$$conf(r) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

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

Horn Rule Mining from KGs

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

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



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

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

AMIE

Language bias: safe and closed rules

safe: every head variable must appear in the body

closed: every variable must appear in at least two atoms

AMIE

Language bias: safe and closed rules

safe: every head variable must appear in the body

closed: every variable must appear in at least two atoms

Algorithm steps:

- maintain a rule queue, starting from an empty rule
- for each rule:
  1. process the rule
     - compute statistics: *supp*, $conf_{PCA}$
     - filter rules based on statistics and output rule
  2. extend the queue by applying refinement operators

AMIE

Language bias: safe and closed rules

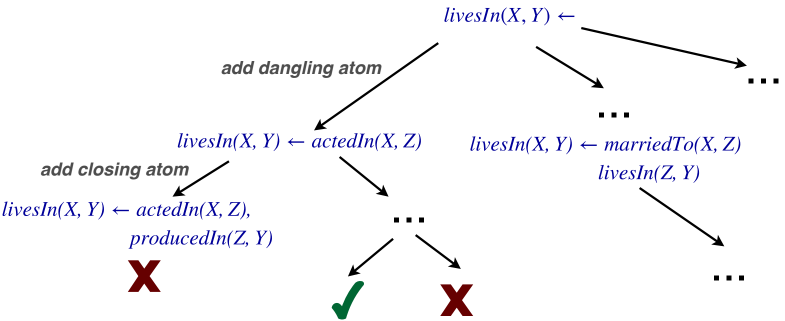
safe: every head variable must appear in the body

closed: every variable must appear in at least two atoms

Algorithm steps:

- maintain a rule queue, starting from an empty rule
- for each rule:
  1. process the rule
     - compute statistics: $supp$, $conf_{PCA}$
     - filter rules based on statistics and output rule
  2. extend the queue by applying refinement operators
     - add dangling atom
     - add closing atom
     - add instantiated atom (with constant)

Refinement Operators



Other Related Works

- RDF2Rules [Wang and Li, 2015]
  - Optimized for cycles (even more restricted language bias)

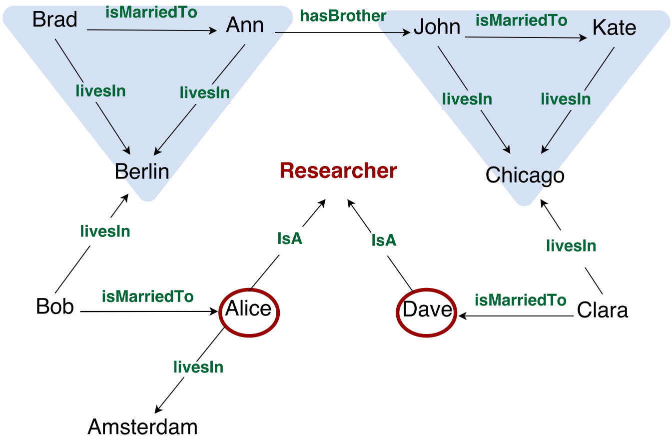
- Ontology path finding [Chen *et al.*, 2016]
  - Parallelizations of the rule evaluation stage

- Comparison of rule measures for KGs [Duc Tran *et al.*, 2018]

- Neural-based rule mining methods [Yang *et al.*, 2017]
  - reduce the rule learning problem to algebraic operations on neural-embedding-based representations of a given KG

Nonmonotonic Rule Mining

Nonmonotonic rule mining from KGs: OWA is a challenge!



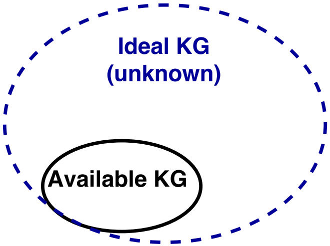
$$r: \ livesIn(X,Z) \leftarrow isMarriedTo(Y,X), livesIn(Y,Z), not \ \ researcher(X)$$

Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG

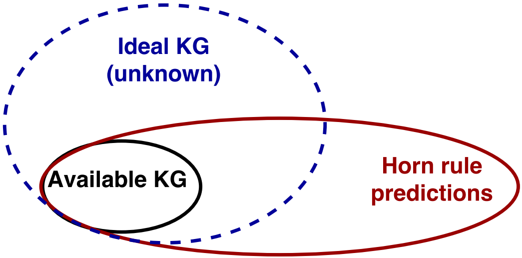


Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set

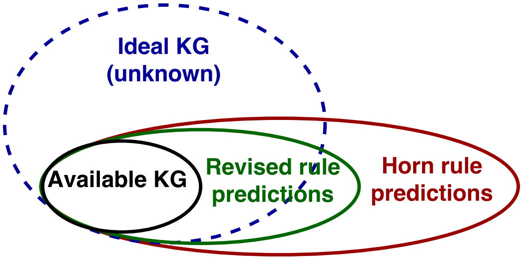


Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set



Find:

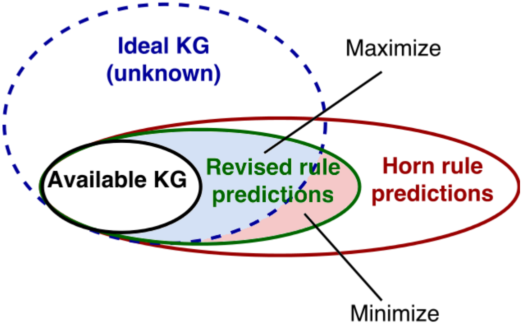
- Nonmonotonic revision of Horn rule set

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?

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

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

Avoid Data Overfitting

How to distinguish exceptions from noise?

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

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

Avoid Data Overfitting

How to distinguish exceptions from noise?

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

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

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

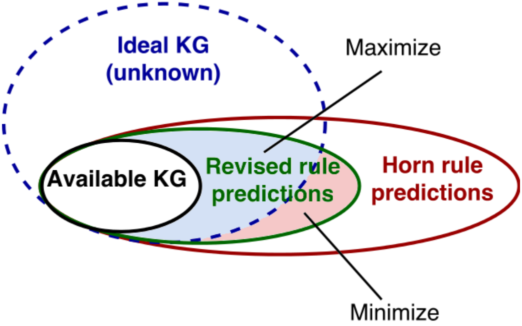
Intuition: Rules with good exceptions should make few conflicting predictions

Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set



Find:

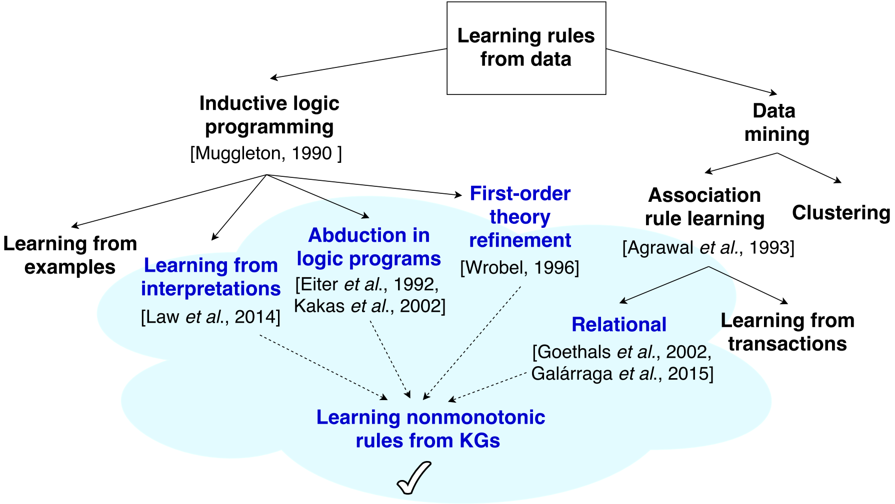
- Nonmonotonic revision of Horn rules, such that
  - number of conflicting predictions is **minimal**
  - average conviction is **maximal**

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*

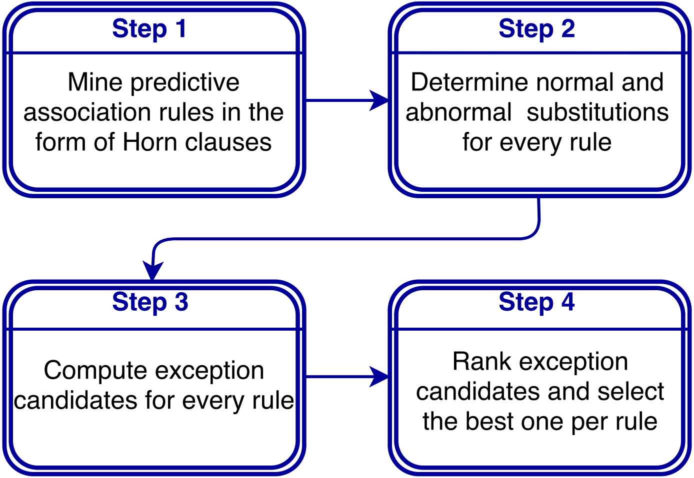
Nonmonotonic Rule Mining from KGs

Goal: learn nonmonotonic rules from KG

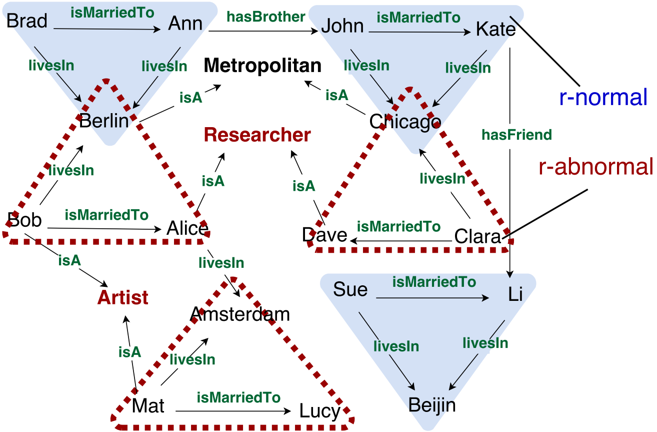
Approach: revise association rules learned using data mining methods



Approach Description



Exception Candidates



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

Exception Ranking

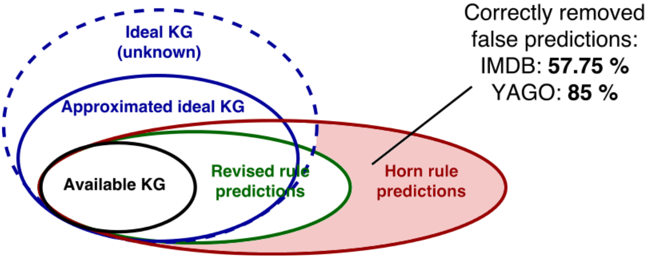
$$\begin{array}{ll} \mathit{rule1} & \{\underline{\mathbf{e_1}}, e_2, e_3, \ldots\} \\ \mathit{rule2} & \{e_1, \underline{\mathbf{e_2}}, e_3, \ldots\} \\ \mathit{rule3} & \{\underline{\mathbf{e_1}}, e_2, e_3, \ldots\} \end{array}$$

Finding globally best revision is expensive, exponentially many candidates!

- Naive ranking: for every rule inject exception that results in the highest conviction
- Partial materialization (PM): apply all rules apart from a given one, inject exception that results in the highest average conviction of the rule and its rewriting
- Ordered PM (OPM): same as PM plus ordered rules application
- Weighted OPM: same as OPM plus weights on predictions

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



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:

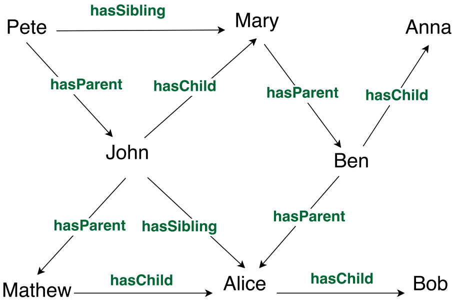
Plots of films in a sequel are written by the same writer, unless a film is American

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

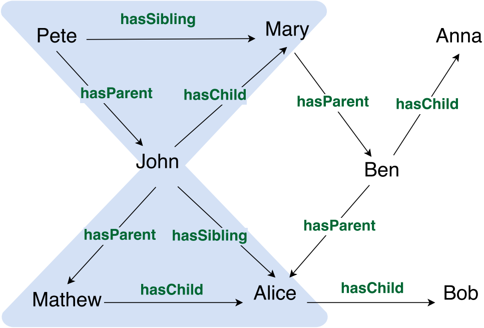
Spouses of film directors appear on the cast, unless they are silent film actors

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

Reasonable Rules

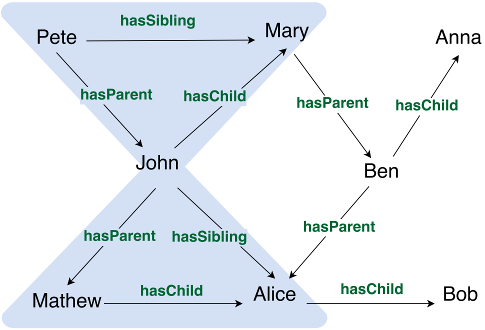


Reasonable Rules



Reasonable Rules

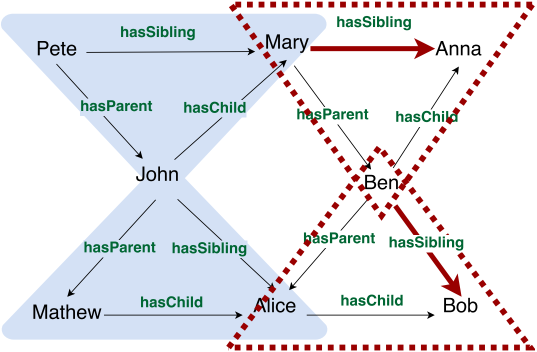
✓ *People with the same parents are likely siblings*



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

Reasonable Rules

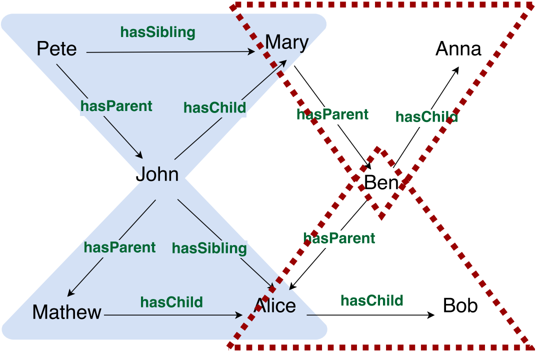
✓ *People with the same parents are likely siblings*



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

Reasonable Rules

✓ *People with the same parents are likely siblings*

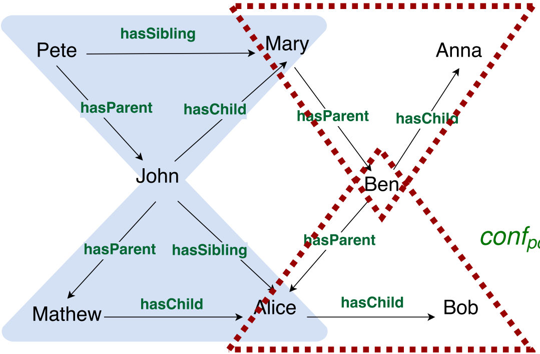


$$conf(r_1) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

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

Reasonable Rules

✓ *People with the same parents are likely siblings*

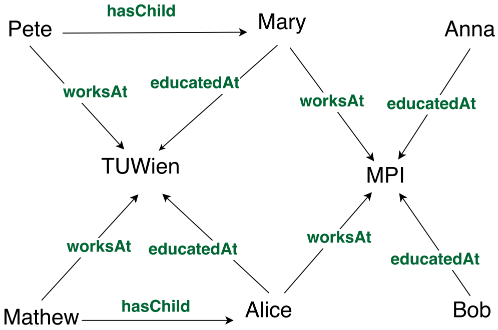


$$conf(r_1) = \frac{|\blacktriangle|}{|\blacktriangle| + |\text{♺}|} = \frac{2}{4}$$

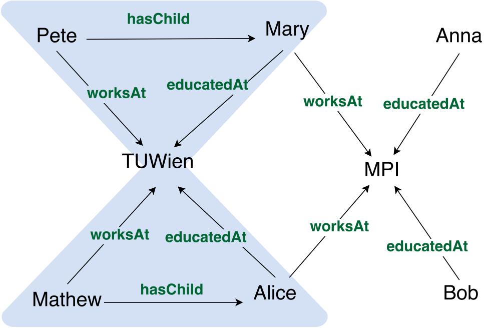
$$conf_{pca}(r_1) = \frac{|\blacktriangle|}{|\{\text{♺} | hasSibling(X, \_) \in \mathcal{G}\}|} = \frac{2}{2}$$

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

Erroneous Rules due to Data Bias

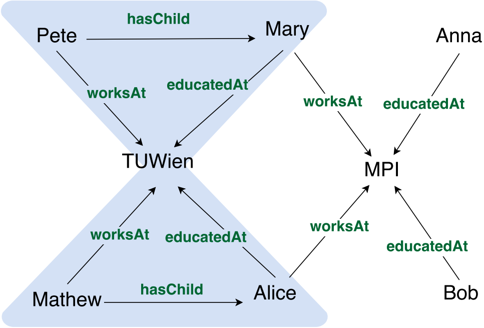


Erroneous Rules due to Data Bias



Erroneous Rules due to Data Bias

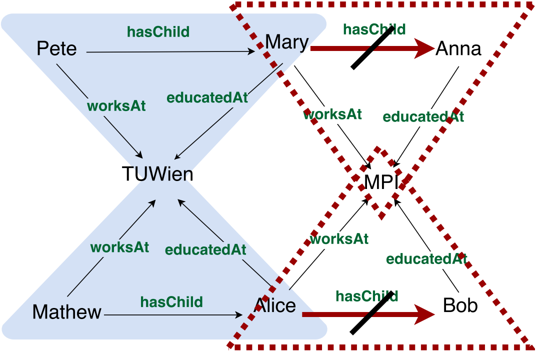
✗ *If one is studying in a university where you teach, he/she is your child*



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

Erroneous Rules due to Data Bias

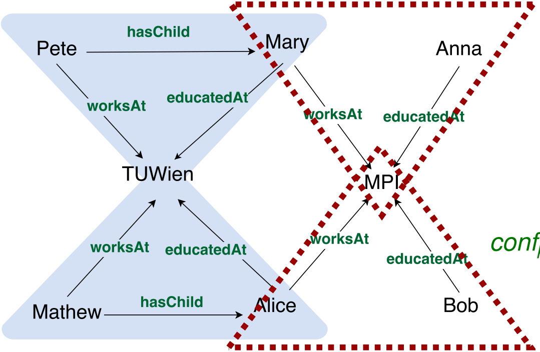
✗ *If one is studying in a university where you teach, he/she is your child*



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

Erroneous Rules due to Data Bias

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



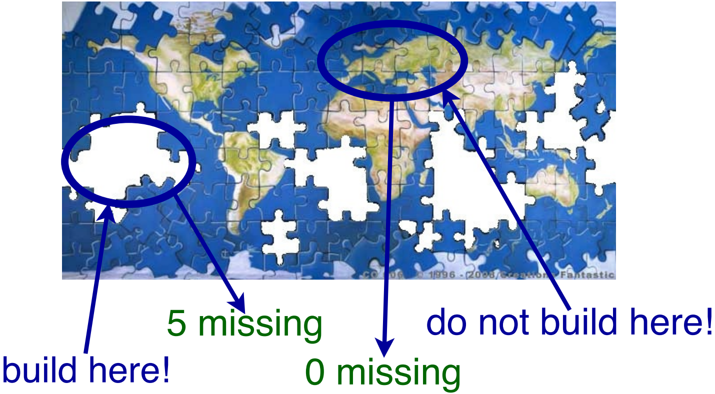
$$conf(r_2) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

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

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

Exploiting Meta-data in Rule Learning

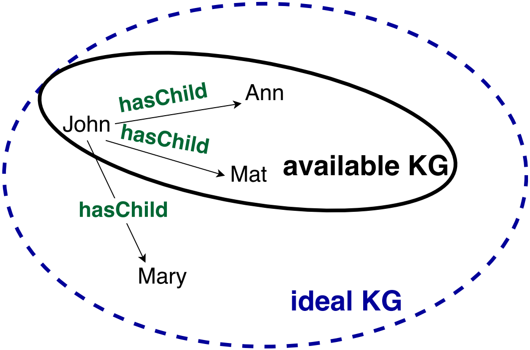
Goal: make use of cardinality constraints on edges of the KG to improve rule learning.



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 $incomplete(p, s)$

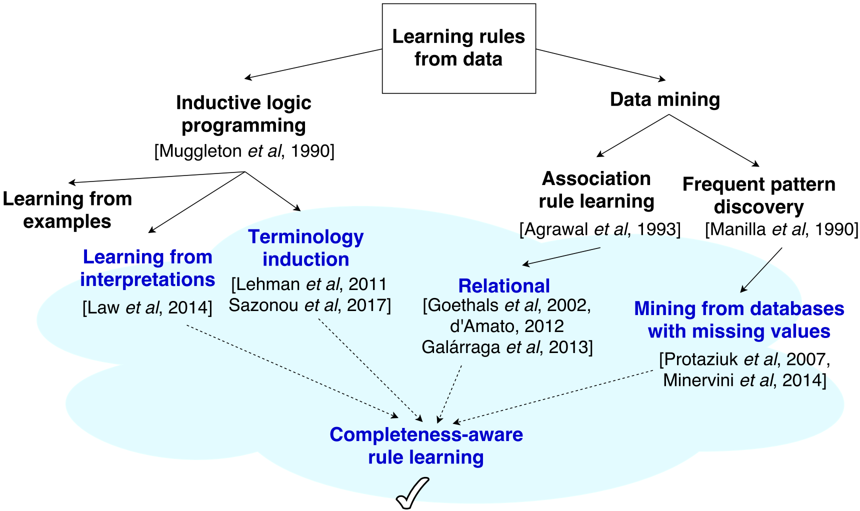


$num(hasChild, john) = 3$
$miss(hasChild, john) = 1$
$incomplete(hasChild, john)$

Cardinality Constraints on Edges

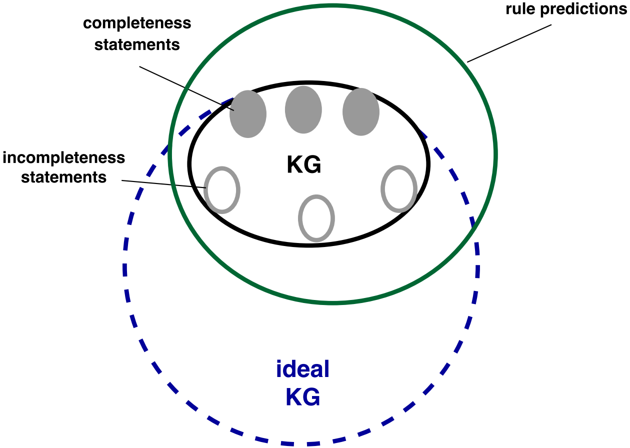
- Mining cardinality assertions from the Web [Mirza *et al.*, 2016]
  - *"... John has **2** children ..."*
- Estimating recall of KGs by crowd sourcing [Razniewski *et al.*, 2016]
  - *20 % of Nobel laureates in physics are missing*
- Predicting completeness in KGs [Galárraga *et al.*, 2017]
  - Add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

Related Work



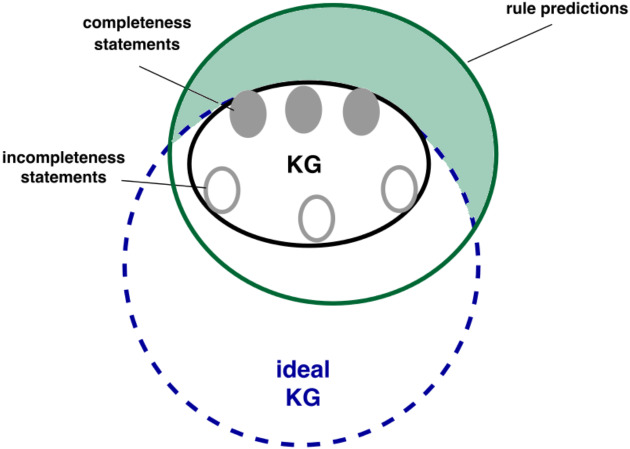
Prediction Post-processing

Remove predictions in complete KG parts [Galárraga *et al.*, 2017], i.e., constraints are set on the output not the input



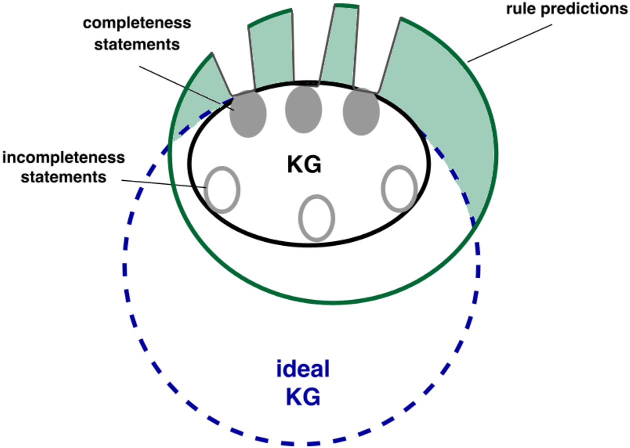
Prediction Post-processing

Remove predictions in complete KG parts [Galárraga *et al.*, 2017], i.e., constraints are set on the output not the input



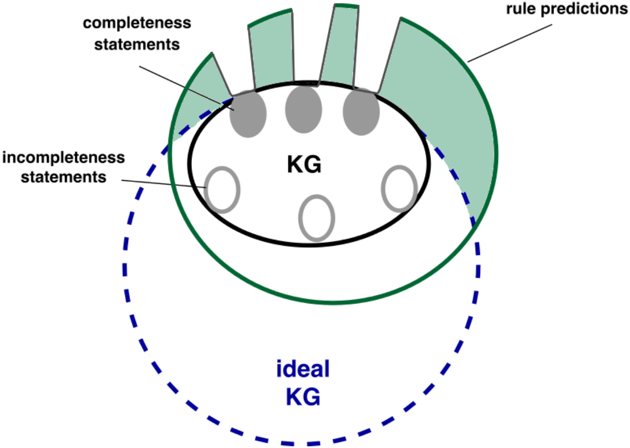
Prediction Post-processing

Remove predictions in complete KG parts [Galárraga *et al.*, 2017], i.e., constraints are set on the output not the input



Prediction Post-processing

Remove predictions in complete KG parts [Galárraga *et al.*, 2017], i.e., constraints are set on the output not the input

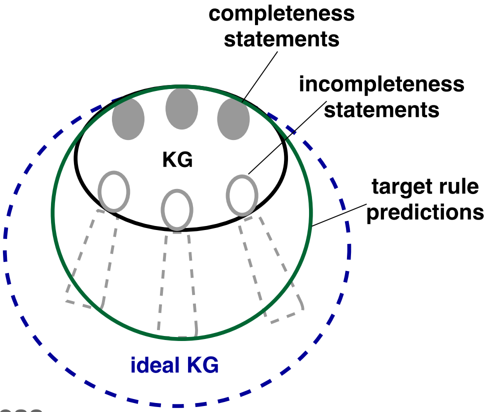


Rules might be still **erroneous**.. What about other incorrect predictions?

Problem Statement

Given:

- KG
- numerical statements



Find: rules which predict

- "few" facts in **complete areas**
- "many" facts in **incomplete areas**

Intuition: rank rules by taking into account numerical constraints on edge counts in the ideal KG

Rule Predictions

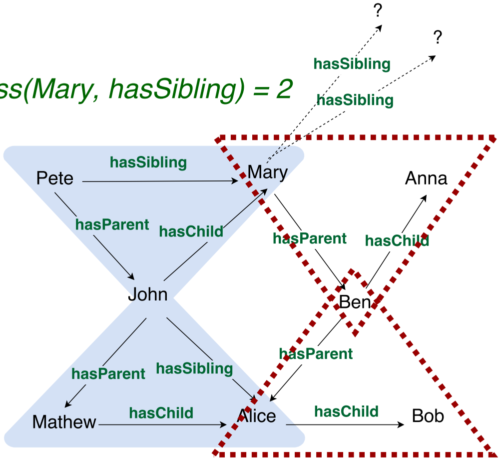
$npi(r)$: number of facts added to incomplete areas by r

$npc(r)$: number of facts added to complete areas by r

Rule Predictions

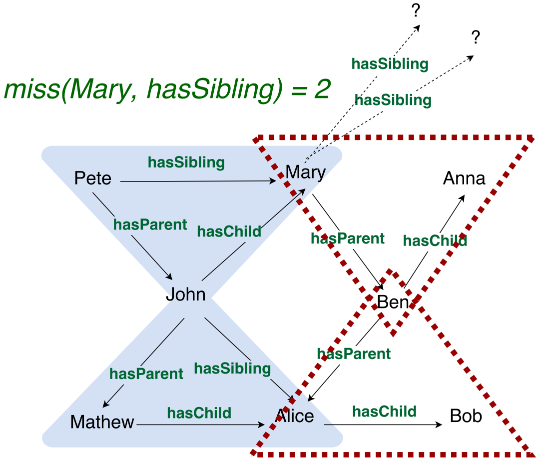
npi(r): number of facts added to incomplete areas by r
npc(r): number of facts added to complete areas by r

miss(Mary, hasSibling) = 2



Rule Predictions

npi(r): number of facts added to incomplete areas by r
npc(r): number of facts added to complete areas by r



$npi(r_1) = 1$
$npc(r_1) = 0$

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

Completeness Confidence

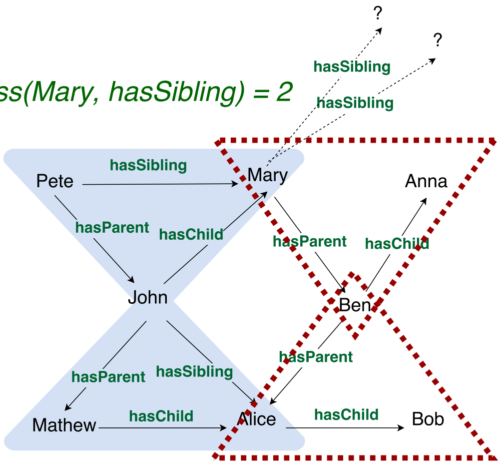
$conf_{comp}$: do not penalize rules that predict new facts in incomplete areas

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

- Generalizes standard confidence ($miss(r) = 0$)
- Generalizes PCA confidence ($miss(r) \in \{0, +\infty\}$)

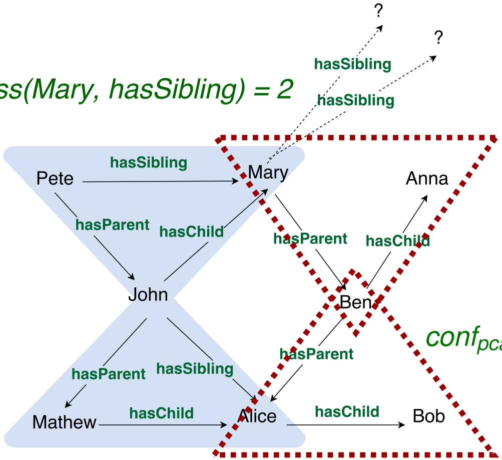
Completeness Confidence Example 1

miss(Mary, hasSibling) = 2



Completeness Confidence Example 1

miss(Mary, hasSibling) = 2



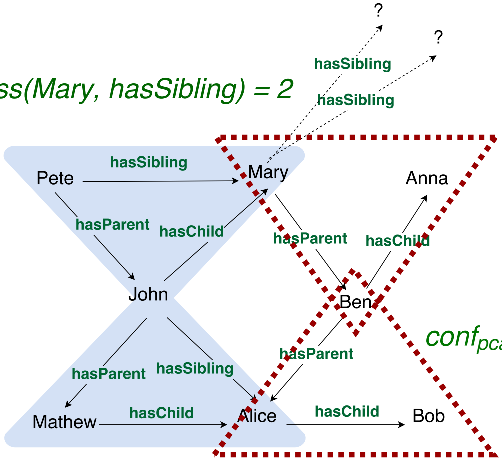
$$conf(r_1) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

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

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

Completeness Confidence Example 1

miss(Mary, hasSibling) = 2



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

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

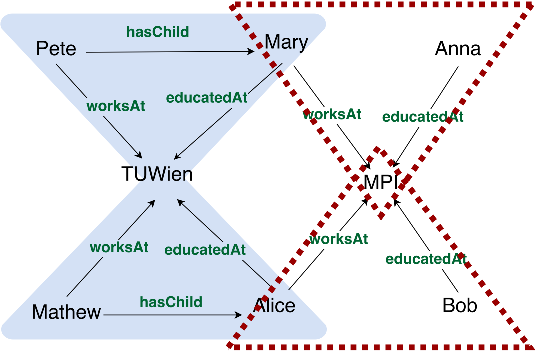
$$npi(r_1) = 1$$

$$conf_{comp}(r_1) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle| - npi(r_1)} = \frac{2}{3}$$

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

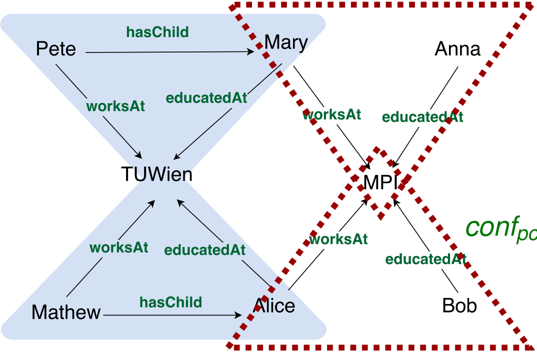
Completeness Confidence Example 2

$miss(hasChild, Alice) = 0$



Completeness Confidence Example 2

$miss(hasChild, Alice) = 0$



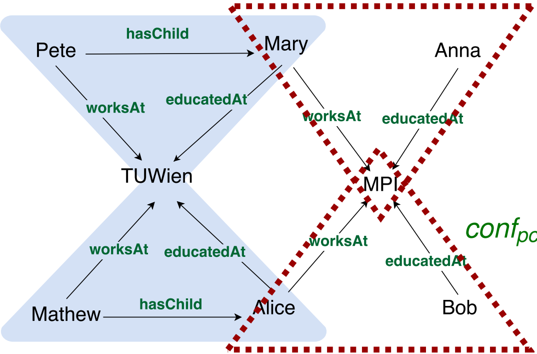
$$conf(r_2) = \frac{|\blacktriangle|}{|\blacktriangle| + |\text{♺}|} = \frac{2}{4}$$

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

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

Completeness Confidence Example 2

$miss(hasChild, Alice) = 0$



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

$$conf_{pca}(r_2) = \frac{|\blacktriangle|}{|\{\triangle | hasChild(Z, \_) \in \mathcal{G}\}|} = \frac{2}{2}$$

$$npi(r_2) = 0$$

$$conf_{comp}(r_2) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle| - npi(r_2)} = \frac{2}{4}$$

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

Other Completeness-aware Measures

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

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

$recall_{comp}$: 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

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

Experimental Setup

2 Datasets:

- WikidataPeople: 2.4M facts over 9 predicates from Wikidata
- LUBM: Synthetic 1.2M facts

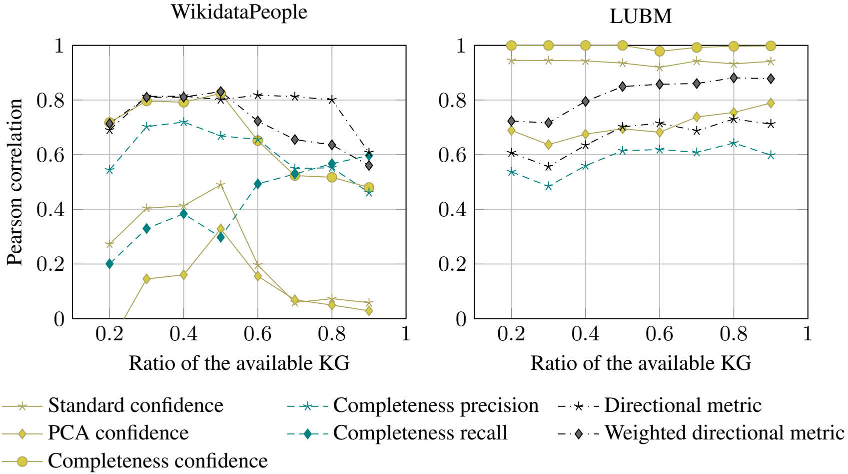
Creation of ideal KG:

- WikidataPeople: using hand made rules
- LUBM: using the OWL ontology

Steps:

- Generate $num(p, x)$ using the ideal KG
- Remove triples randomly to create the available KG
- Mine $r(X, Z) \leftarrow p(X, Y), q(Y, Z)$ rules
- Gold standard: ratio of facts generated in the ideal KG

Experimental Evaluation



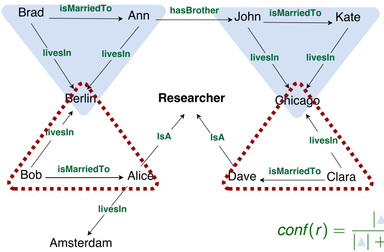
Knowledge Graph Completion

- **Given:** a KG, i.e., set of $\langle s\ p\ o \rangle$ facts and possibly text
- **Find:** missing $\langle s\ p\ o \rangle$ facts

Knowledge Graph Completion

- **Given:** a KG, i.e., set of $\langle s\ p\ o \rangle$ facts and possibly text
- **Find:** missing $\langle s\ p\ o \rangle$ facts

Rule-based approaches



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

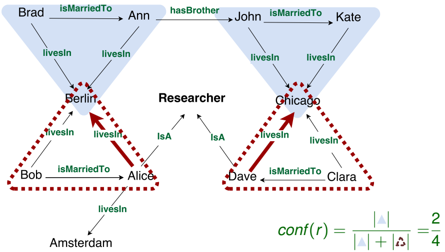
$r:\ livesIn(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z)$

AMIE [Galarraga *et al.*, 2015],
RUMIS [Tran *et al.*, 2016], CARL [Tanon *et al.*, 2017], etc.

Knowledge Graph Completion

- **Given:** a KG, i.e., set of $\langle s\ p\ o\rangle$ facts and possibly text
- **Find:** missing $\langle s\ p\ o\rangle$ facts

Rule-based approaches



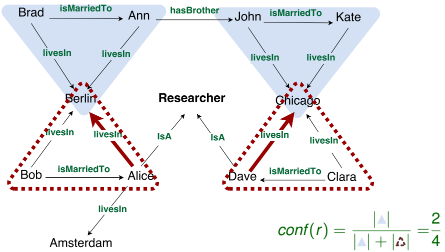
$r: \ livesIn(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z)$

AMIE [Galarraga *et al.*, 2015],
RUMIS [Tran *et al.*, 2016], CARL [Tanon *et al.*, 2017], etc.

Knowledge Graph Completion

- **Given:** a KG, i.e., set of $\langle s\ p\ o\rangle$ facts and possibly text
- **Find:** missing $\langle s\ p\ o\rangle$ facts

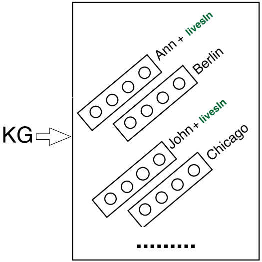
Rule-based approaches



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

AMIE [Galarraga *et al.*, 2015],
RUMIS [Tran *et al.*, 2016], CARL [Tanon *et al.*, 2017], etc.

Statistics-based approaches

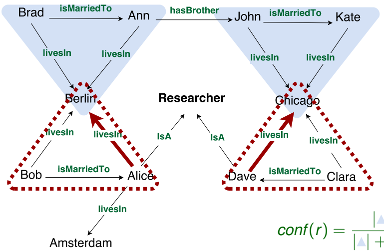


TransE [Bordes *et al.*, 2013], TEKE [Wang and Li, 2016],
RESCAL [Nickel *et al.*, 2011], etc.

Knowledge Graph Completion

- **Given:** a KG, i.e., set of $\langle s\ p\ o \rangle$ facts and possibly text
- **Find:** missing $\langle s\ p\ o \rangle$ facts

Rule-based approaches

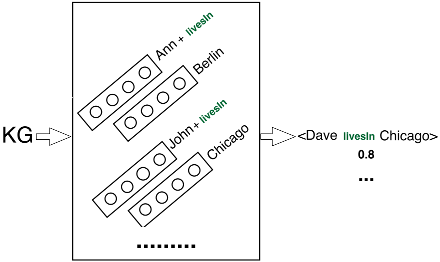


$$conf(r) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

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

AMIE [Galarraga *et al.*, 2015],
RUMIS [Tran *et al.*, 2016], CARL [Tanon *et al.*, 2017], etc.

Statistics-based approaches

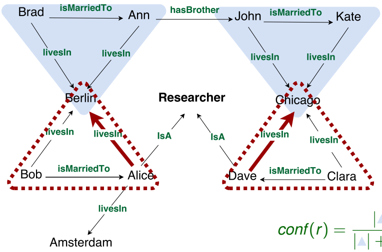


TransE [Bordes *et al.*, 2013], TEKE [Wang and Li, 2016],
RESCAL [Nickel *et al.*, 2011], etc.

Knowledge Graph Completion

- **Given:** a KG, i.e., set of $\langle s\ p\ o\rangle$ facts and possibly text
- **Find:** missing $\langle s\ p\ o\rangle$ facts

Rule-based approaches

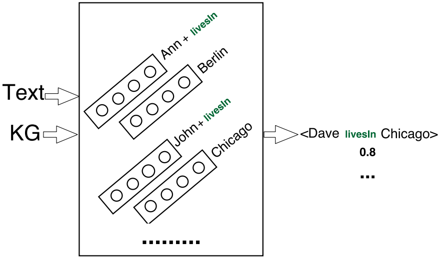


$$conf(r) = \frac{|\triangle|}{|\triangle| + |\triangle|} = \frac{2}{4}$$

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

AMIE [Galarraga *et al.*, 2015],
RUMIS [Tran *et al.*, 2016], CARL [Tanon *et al.*, 2017], etc.

Statistics-based approaches



TransE [Bordes *et al.*, 2013], TEKE [Wang and Li, 2016],
RESCAL [Nickel *et al.*, 2011], etc.

Motivation

Goal: Combine available techniques into a hybrid method

Rule-based approaches

+ Interpretable
+ Limited training data
- Local patterns
- Not extendable

Statistics-based approaches

- Hard to interpret
- A lot of training data
+ Global patterns
+ Extandable (e.g., text)

Proposed solution

Precompute KG embedding and treat the result as an oracle, which can be queried any time during rule construction.

T. Vinh Ho, D. Stepanova, M. Gad-Elrab, E. Kharlamov, G. Weikum. Rule Learning from KGs guided by Embedding Models. *ISWC2018*.

Problem Statement

Feedback-driven rule mining

- **Given:**
  - KG
  - Embedding model
  - Type of rules to be learned (e.g., with(out) negation, disjunctive, etc.)
- **Find:**
  - a set of rules of the desired type, which agree with embedding model on predictions that they make

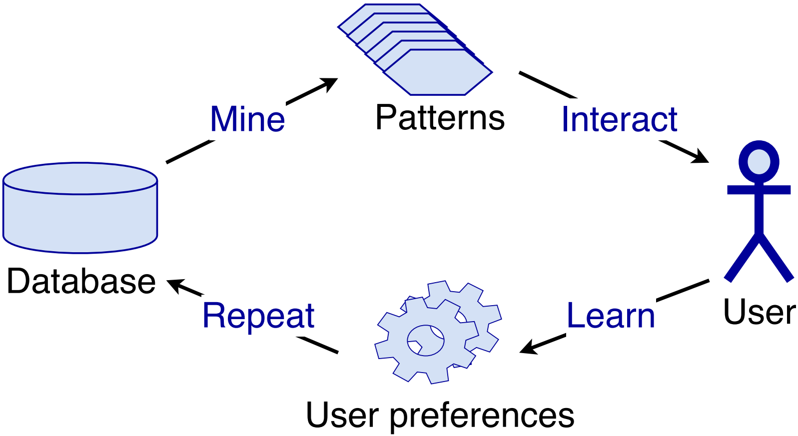
Related Work

- **Constraints in embedding models**
  - Injecting logical formulas as constraints into embedding models (output is still a set of predictions; unclear where they came from) [Guo *et al.*, 2017]

- **Rule mining with external support**
  - Interactive pattern mining [Goethals *et al.*, 2011], [Dzyuba and van Leeuwen, 2017]
  - Interactive association rule mining [Skrabal *et al.*, 2012]

Mine-Interact-Learn-Repeat

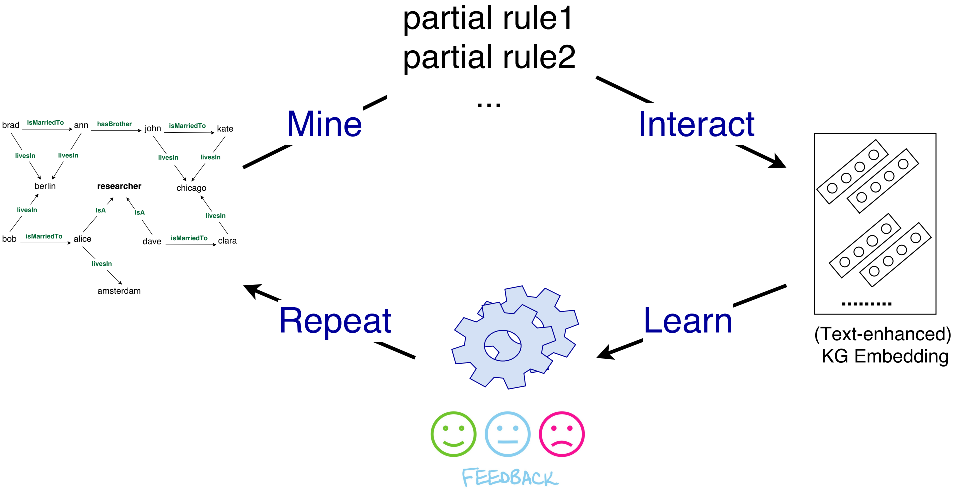
Mimic "mine-interact-learn-repeat" schema [Dzyuba and van Leeuwen, 2017]



Mine-Interact-Learn-Repeat

Mimic "mine-interact-learn-repeat" schema [Dzyuba and van Leeuwen, 2017]

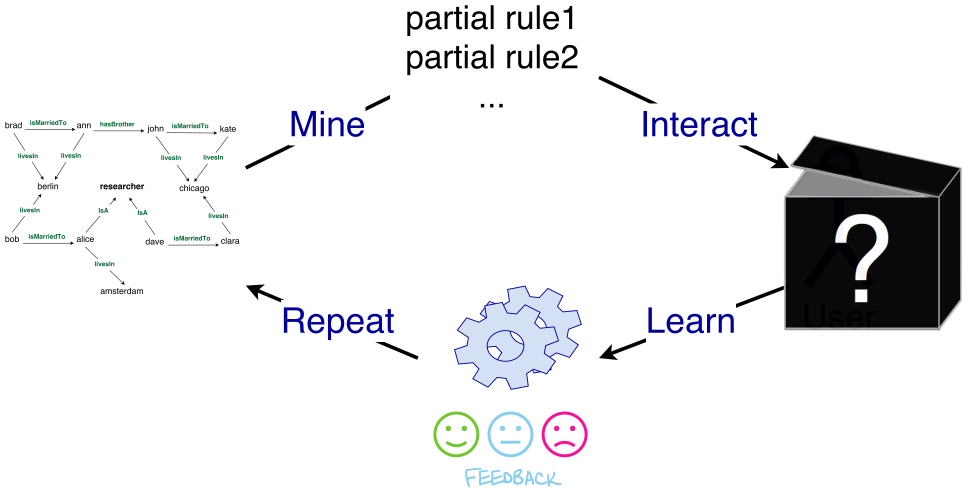
Establish "user-in-the-loop" inspired interaction between the rule mining algorithm and the embedding model



Mine-Interact-Learn-Repeat

Mimic "mine-interact-learn-repeat" schema [Dzyuba and van Leeuwen, 2017]

Establish "user-in-the-loop" inspired interaction between the rule mining algorithm and the embedding model



Research Questions

Q1 **(Interact)** What kind of feedback is required/possible to obtain from the “black box” to organize convenient and effective interaction process?

Research Questions

Q1 **(Interact)** What kind of feedback is required/possible to obtain from the “black box” to organize convenient and effective interaction process?

Q2 **(Mine)** How to adapt existing rule mining algorithms to account for feedback?

Research Questions

Q1 **(Interact)** What kind of feedback is required/possible to obtain from the "black box" to organize convenient and effective interaction process?

Q2 **(Mine)** How to adapt existing rule mining algorithms to account for feedback?

Q3 **(Learn)** Can anything be learnt from the feedback provided by embeddings?

Research Questions

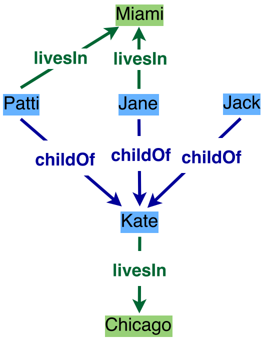
Q1 **(Interact)** What kind of feedback is required/possible to obtain from the "black box" to organize convenient and effective interaction process?

Q2 **(Mine)** How to adapt existing rule mining algorithms to account for feedback?

Q3 **(Learn)** Can anything be learnt from the feedback provided by embeddings?

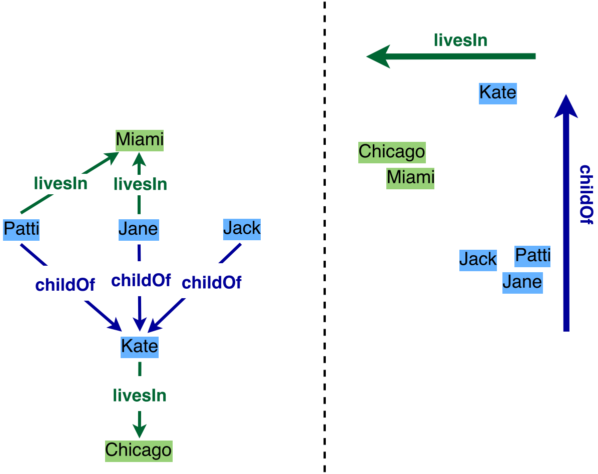
Embedding-based Methods

- **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



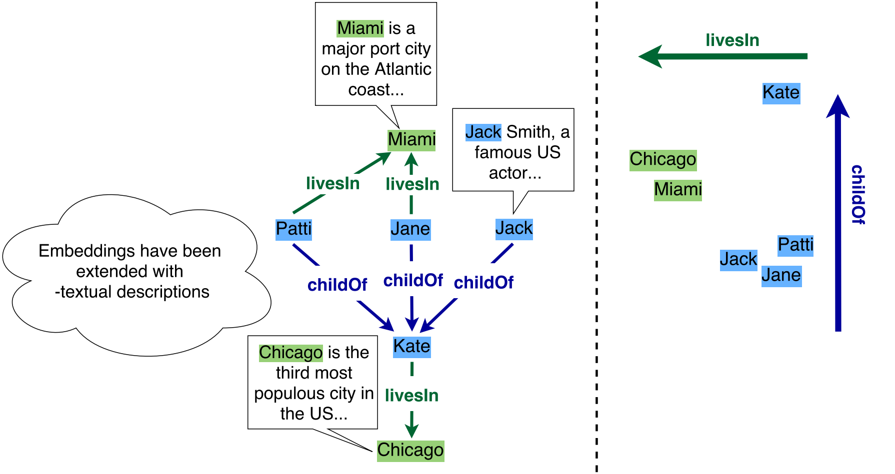
Embedding-based Methods

- **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



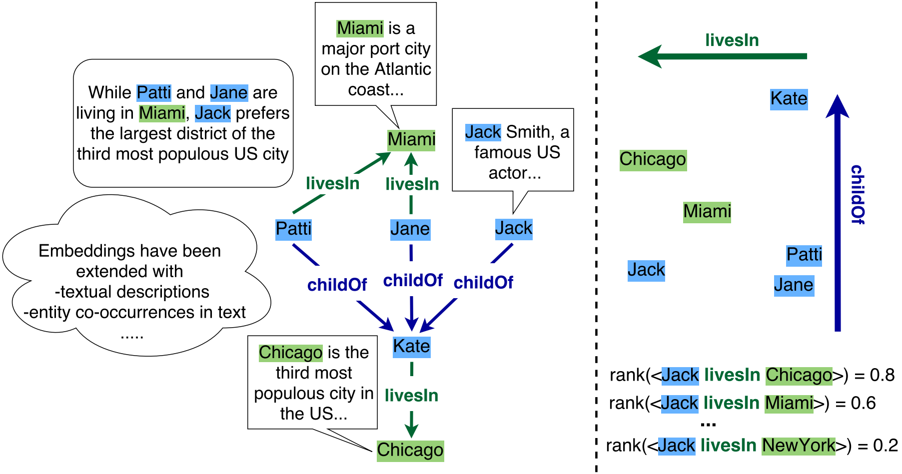
Embedding-based Methods

- **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



Embedding-based Methods

- **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



Q1 (Interact)

Measure quality of $r : p(X, Y) \leftarrow B$, based on the embedding model

- rely on average quality of predicted facts

$$rule\_mrr(r) = \frac{1}{|predictions(r)|} \sum_{<s\,p\,o> \in predictions(r)} rank(<s\,p\,o>)$$

Q1 (Interact)

Measure quality of $r : p(X, Y) \leftarrow B$, based on the embedding model

- rely on average quality of predicted facts

$$rule\_mrr(r) = \frac{1}{|predictions(r)|} \sum_{<s\,p\,o>\in predictions(r)} rank(<s\,p\,o>)$$

Example

$$livesIn(X, Y) \leftarrow actedIn(X, Z), producedIn(Z, Y)$$

- rule predictions: $<Jack\ livesIn\ NY>$, $<Mat\ livesIn\ Berlin>$

$$rule\_mrr(r) = \frac{rank(<\text{Jack}\ \textbf{livesIn}\ \text{NY}>) + rank(<\text{Mat}\ \textbf{livesIn}\ \text{Berlin}>)}{2}$$

(Q1) Interact

Measure quality of $r : h(X, Y) \leftarrow B$, based on the embedding model

- rely on average quality of predicted facts estimated by embeddings

$$rule\_mrr(r) = \frac{1}{|N|} \sum_{s,h,o \in N} \frac{1}{rank(s, h, o)}$$

- combination of mrr with standard rule measures over KG

$$embed\_conf(r) = \lambda * conf(r) + (1 - \lambda) * rule\_mrr(r),$$

(Q1) Interact

Measure quality of $r : h(X, Y) \leftarrow B$, based on the embedding model

- rely on average quality of predicted facts estimated by embeddings

$$rule\_mrr(r) = \frac{1}{|N|} \sum_{s,h,o \in N} \frac{1}{rank(s, h, o)}$$

- combination of mrr with standard rule measures over KG

$$embed\_conf(r) = \lambda * conf(r) + (1 - \lambda) * rule\_mrr(r),$$

  - λ: a weighting factor
  - *conf*: descriptive quality based on the original KG
    any other standard rule measure can be plugged in
  - *rule_mrr*: predictive quality based on KG embedding
    any embedding model including text-enhanced ones can be used

- more complex interaction, e.g., information theoretic measures?

Research Questions

Q1 **(Interact)** What kind of feedback is required/possible to obtain from the "black box" to organize convenient and effective interaction process?

Q2 **(Mine)** How to adapt existing rule mining algorithms to account for feedback?

Q3 **(Learn)** Can anything be learnt from the feedback provided by embeddings?

(Q2) Mine

Algorithm steps:

- maintain a rule queue, starting from an empty rule
- for each rule:
  1. process the rule
  2. extend the queue by applying refinement operators

(Q2) Mine

Algorithm steps:

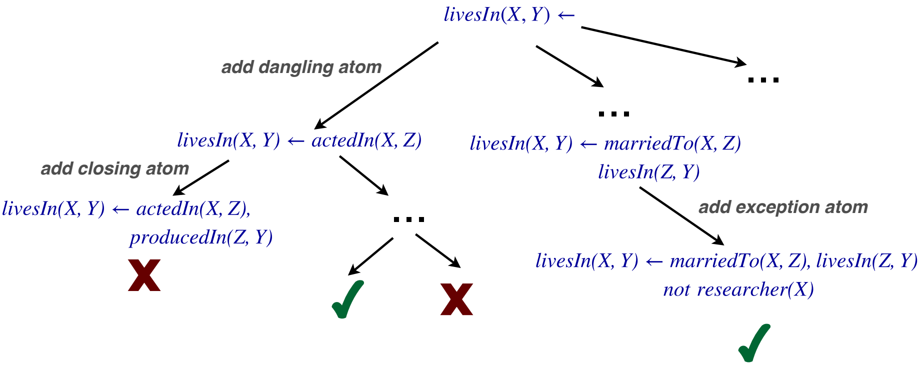
- maintain a rule queue, starting from an empty rule
- for each rule:
  1. process the rule
     - compute statistics: *rule_mrr*, *embed_conf*...
     - filter rules based on statistics and output rule
  2. extend the queue by applying refinement operators

(Q2) Mine

Algorithm steps:

- maintain a rule queue, starting from an empty rule
- for each rule:
  1. process the rule
     - compute statistics: *rule_mrr*, *embed_conf*...
     - filter rules based on statistics and output rule
  2. extend the queue by applying refinement operators
     - add dangling atom
     - add closing atom
     - add positive unary atom
     - add exception unary atom
     - add exception binary atom

Refinement Operators



- Exploit embedding to prune rule search space
- Generate rule language bias dynamically

Open Questions

Q1 **(Interact)** What kind of feedback is required/possible to obtain from the "black box" to organize convenient and effective interaction process?

Q2 **(Mine)** How to adapt existing rule mining algorithms to account for feedback?

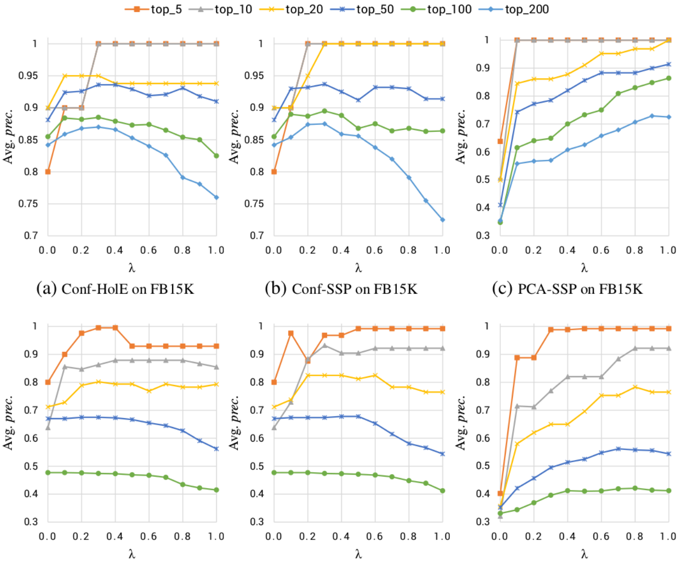
Q3 **(Learn)** Can anything be learnt from the feedback provided by embeddings?

- Ideally, we want to learn the structure of most promising rules, i.e., the best rules have at most 5 atoms, 4 variables, etc..

Experimental Setup

- Datasets:
  - FB15K: 592M facts, 15M entities and 1345 relations relations
  - Wiki44K: 250M facts, 44M entities and 100 relations
- Ideal graph: remove 80% of facts for every relation
- Embedding models: TransE, HolE, SSP
- For every dataset selected a model that works best
  - Evaluate predictive capabilities of rules obtained by our system vs others

Evaluation



(a) Conf-HolE on FB15K (b) Conf-SSP on FB15K (c) PCA-SSP on FB15K

(d) Conf-TransE on Wiki44K (e) Conf-SSP on Wiki44K (f) PCA-SSP on Wiki44K

Example Rules

Examples of rules learned from Wikidata

By default uni graduates are nationals of the country where the uni is located, but not in the case of research institutions

r_1 : $\mathit{nationality}(X, Y) \leftarrow \mathit{graduatedFrom}(X, Z), \mathit{inCountry}(Z, Y), \mathit{not\ researchUni}(Z)$

Script writers stay the same across sequels, but not for TV series

r_2 : $\mathit{scriptwriterOf}(X, Y) \leftarrow \mathit{preceededBy}(X, Z), \mathit{scriptWriterOf}(Y, Z), \mathit{not\ tvSeries}(Z)$

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

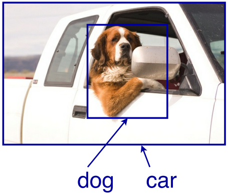
r_3 : $\mathit{nobleFamily}(X, Y) \leftarrow \mathit{spouse}(X, Z), \mathit{nobleFamily}(Z, Y), \mathit{not\ chineseDynasty}(Y)$

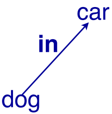
Commonsense Knowledge

"AI has seen great advances of many kinds recently, but there is one critical area where progress has been extremely slow: ordinary commonsense." [Davis and Marcus, 2015]

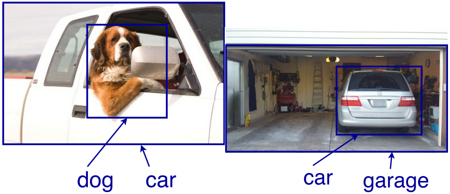
- Questions that are easy for people but hard for machines
  - "Who is **taller**, Prince William or **his baby son** Prince George?"
  - "Can you make a **salad** out of a **polyester shirt**?"
  - "Can an **elephant** sit on a **tree**?"

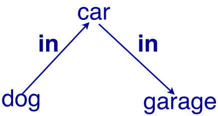
Commonsense Rule Induction from Hybrid Sources



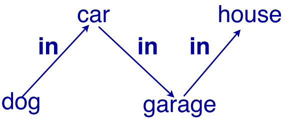


Commonsense Rule Induction from Hybrid Sources

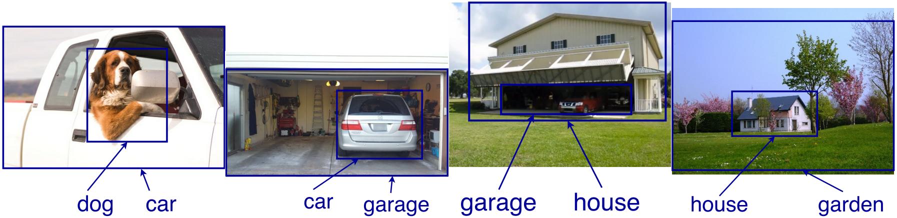


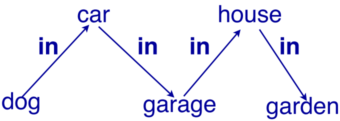


Commonsense Rule Induction from Hybrid Sources

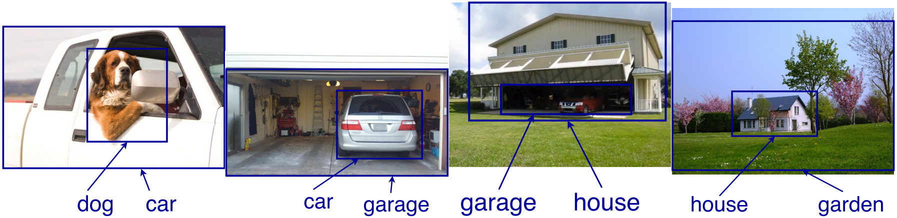


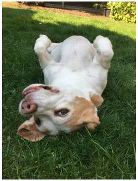
Commonsense Rule Induction from Hybrid Sources



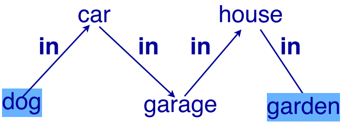


Commonsense Rule Induction from Hybrid Sources

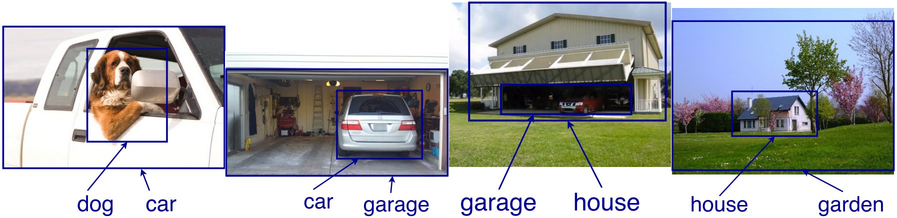


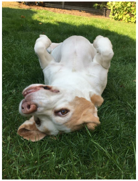


Dog enjoying the sun in the garden

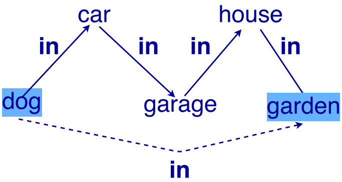


Commonsense Rule Induction from Hybrid Sources

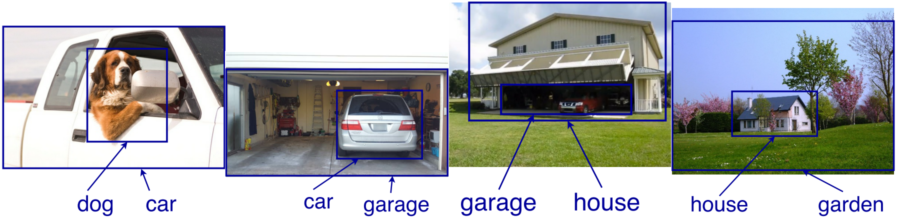


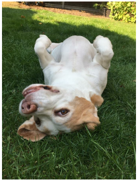


Dog enjoying the sun in the garden

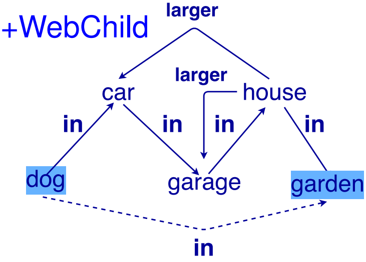


Commonsense Rule Induction from Hybrid Sources



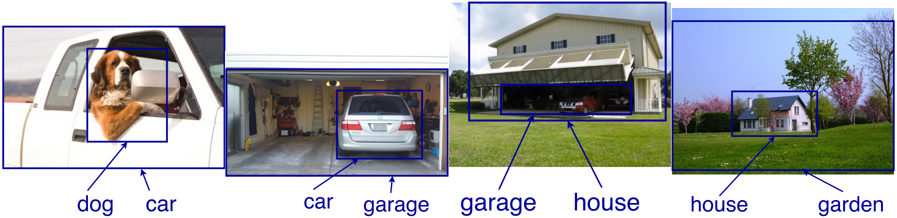


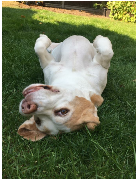
Dog enjoying the sun in the garden



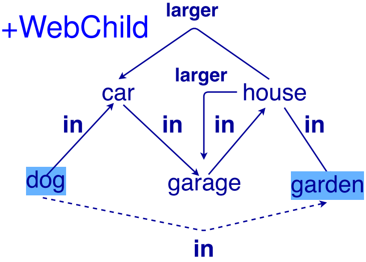
WebChild KG [Tandon *et al.*, 2017]

Commonsense Rule Induction from Hybrid Sources





Dog enjoying the sun in the garden



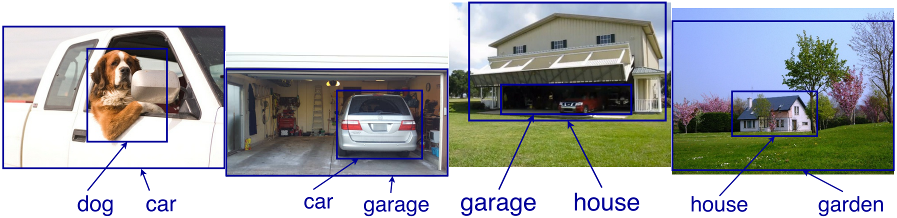
Desired output:

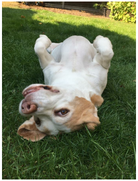
$larger(Y, X) \leftarrow in(X, Y)$
$heavier(Y, X) \leftarrow on(X, Y)$
$has(X, wings) \vee round(X) \leftarrow in(X, sky)$

...

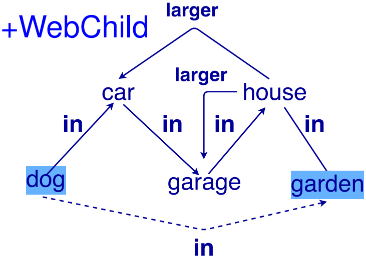
WebChild KG [Tandon *et al.*, 2017]

Commonsense Rule Induction from Hybrid Sources





Dog enjoying the sun in the garden



Desired output:

$larger(Y, X) \leftarrow in(X, Y)$
$heavier(Y, X) \leftarrow on(X, Y)$
$has(X, wings) \vee round(X) \leftarrow in(X, sky)$

$\ldots$

WebChild KG [Tandon *et al.*, 2017]
Reasoning over images: [Eiter and Kaminski, 2016], [Donadello *et al.*, 2017], etc.

Commonsense Rules from Text

- SHERLOCK [Schoenmackers *et al.*, 2010]: Early attempt to learn rules from open domain text extractions.

- [Gordon and Schubert, 2011]: Utilizes presuppositional discourse patterns (such as statements with but, yet ... etc) to collect conditional knowledge in the form of *if-then* rules.

- [Petrova and Rudolph, 2016]: Rules from consessional statements "Although he is a researcher, he never moved." leads to a rule "Researchers normally move frequently."

- [Dragoni *et al.*, 2016]: Rules from legal documents

- KG + text?

Other Rule Types to Consider

- **Disjunctive:**
  $male(Y) \vee female(Y) \leftarrow hasParent(X, Y)$

Other Rule Types to Consider

- **Disjunctive:**
  $male(Y) \vee female(Y) \leftarrow hasParent(X, Y)$
- **Existential:**
  $\exists Y\ hasParent(X, Y) \leftarrow person(X)$

Other Rule Types to Consider

- **Disjunctive:**
  $male(Y) \vee female(Y) \leftarrow hasParent(X, Y)$
- **Existential:**
  $\exists Y\ hasParent(X, Y) \leftarrow person(X)$
- **Constraints:**
  $\perp \leftarrow hasParent(X, Y), hasParent(Y, X)$

Other Rule Types to Consider

- **Disjunctive:**
  $male(Y) \vee female(Y) \leftarrow hasParent(X, Y)$
- **Existential:**
  $\exists Y\ hasParent(X, Y) \leftarrow person(X)$
- **Constraints:**
  $\perp \leftarrow hasParent(X, Y), hasParent(Y, X)$
- **Temporal constraints:**
  $\perp \leftarrow bornIn(X, Y), after(Y, Z), studied(X, Z)$

Outlook Issues

- Rules from hybrid sources
- Complex rule types, e.g., numerical, constraints, datalog+-
- Background knowledge
- Causality and novel rule measures
- Exploit external functions possibly as a blackbox
- Rule learning from commonsense KGs
- Optimizations

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, Luis Gallaraga

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.

Yang Chen, Sean Louis Goldberg, Daisy Zhe Wang, and Soumitra Siddharth Johri.
Ontological pathfinding.
In *SIGMOD*, pages 835–846. ACM, 2016.

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.

Ernest Davis and Gary Marcus.
Commonsense reasoning and commonsense knowledge in artificial intelligence.
Commun. ACM, 58(9):92–103, 2015.

Ivan Donadello, Luciano Serafini, and Artur S. d'Avila Garcez.
Logic tensor networks for semantic image interpretation.
In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1596–1602, 2017.

Mauro Dragoni, Serena Villata, Williams Rizzi, and Guido Governatori.
Combining nlp approaches for rule extraction from legal documents.
In *1st Workshop on MIning and REasoning with Legal texts (MIREL)*, 2016.

Minh Duc Tran, Claudia d'Amato, Binh Thanh Nguyen, and Andrea G. B. Tettamanzi.
Comparing rule evaluation metrics for the evolutionary discovery of multi-relational association rules in the semantic web.
In *Genetic Programming*, pages 289–305, 2018.

Vladimir Dzyuba and Matthijs van Leeuwen.
Learning what matters - sampling interesting patterns.
In *PAKDD (1)*, volume 10234 of *Lecture Notes in Computer Science*, pages 534–546, 2017.

References II

Thomas Eiter and Tobias Kaminski.
Exploiting contextual knowledge for hybrid classification of visual objects.
In *Logics in Artificial Intelligence - 15th European Conference, JELIA 2016, Larnaca, Cyprus, November 9-11, 2016, Proceedings*, pages 223–239, 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.

Bart Goethals, Sandy Moens, and Jilles Vreeken.
MIME: a framework for interactive visual pattern mining.
In *KDD*, pages 757–760. ACM, 2011.

Jonathan Gordon and Lenhart K. Schubert.
Discovering commonsense entailment rules implicit in sentences.
In *TextInfer Workshop on Textual Entailment*, TIWTE '11, pages 59–63, 2011.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo.
Knowledge graph embedding with iterative guidance from soft rules.
CoRR, 2017.

References III

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-A. Krogel, Simon Alan Rawles, Filip Zelezný, Peter A. Flach, Nada Lavrac, and Stefan Wrobel.
Comparative evaluation of approaches to propositionalization.
In *ILP*, pages 197–214, 2003.

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

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.

References IV

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel.
A three-way model for collective learning on multi-relational data.
In *ICML*, 2011.

Alina Petrova and Sebastian Rudolph.
Web-mining defeasible knowledge from concessional statements.
In *Graph-Based Representation and Reasoning - 22nd International Conference on Conceptual Structures, ICCS 2016, Annecy, France, July 5-7, 2016, Proceedings*, pages 191–203, 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.

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

Stefan Schoenmackers, Oren Etzioni, Daniel S. Weld, and Jesse Davis.
Learning first-order horn clauses from web text.
In *EMNLP*, pages 1088–1098, 2010.

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

Radek Skrabal, Milan Simunek, Stanislav Vojír, Andrej Hazucha, Tomás Marek, David Chudán, and Tomás Kliegr.
Association rule mining following the web search paradigm.
In *ECML/PKDD (2)*, volume 7524 of *Lecture Notes in Computer Science*, pages 808–811. Springer, 2012.

References V

- Niket Tandon, Gerard de Melo, and Gerhard Weikum.
  Webchild 2.0 : Fine-grained commonsense knowledge distillation.
  In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, System Demonstrations*, pages 115–120, 2017.
- Thomas Pellissier Tanon, Daria Stepanova, Simon Razniewski, Paramita Mirza, and Gerhard Weikum.
  Completeness-aware rule learning from knowledge graphs.
  In *ISWC*, pages 507–525, 2017.
- Hai Dang Tran, Daria Stepanova, Mohamed H. Gad-Elrab, Francesca A. Lisi, and Gerhard Weikum.
  Towards nonmonotonic relational learning from knowledge graphs.
  In *ILP*, pages 94–107, 2016.
- Zhichun Wang and Juan-Zi Li.
  Rdf2rules: Learning rules from RDF knowledge bases by mining frequent predicate cycles.
  CoRR, abs/1512.07734, 2015.
- Zhigang Wang and Juan-Zi Li.
  Text-enhanced representation learning for knowledge graph.
  In *IJCAI*, 2016.
- Fan Yang, Zhilin Yang, and William W. Cohen.
  Differentiable learning of logical rules for knowledge base reasoning.
  In *NIPS*, pages 2316–2325, 2017.