

Rule Induction and Reasoning over Knowledge Graphs

Daria Stepanova^{1,2}, Mohamed Gad-Elrab¹, Vinh Thinh Ho¹

¹Max Planck Institute for Informatics, Saarbrücken, Germany

²Bosch Center for Artificial Intelligence, Renningen, Germany

25.09.2018



max planck institut
informatik



BOSCH
Invented for life



Outline

Motivation

Preliminaries

Horn Rules

Nonmonotonic Rules

Rule Learning

ILP

KG Challenges

Association Rules

Relational Association Rules

Exception-awareness

Incompleteness

Rules from Hybrid Sources

Further Topics

Motivation

Preliminaries

Rule Learning

Exception-awareness

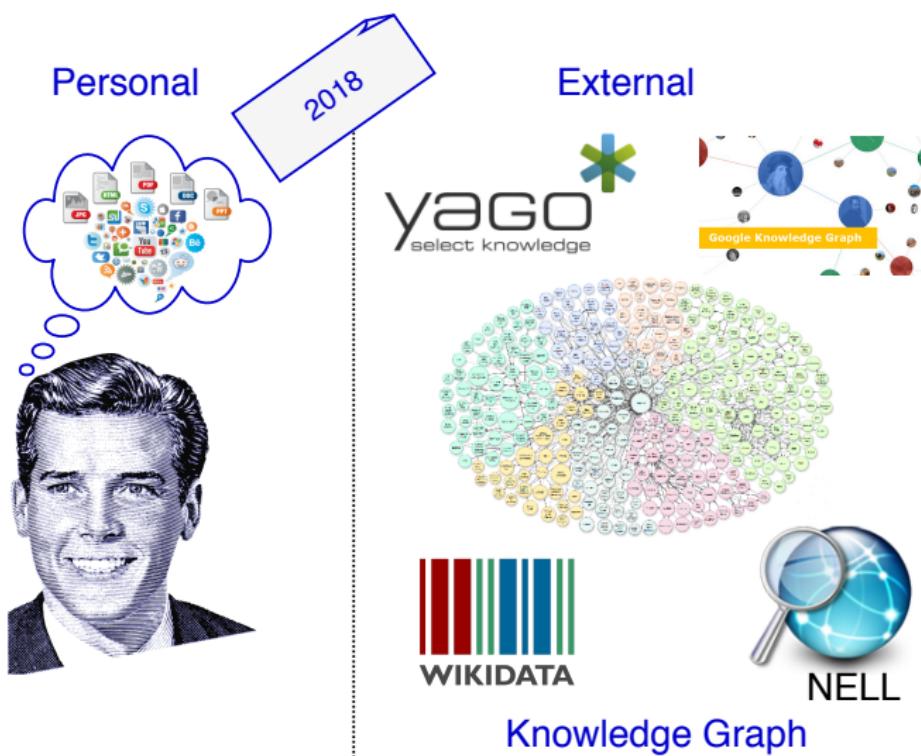
Incompleteness

Rules from Hybrid Sources

Further Topics

Knowledge Graphs

"Semantically enriched machine processable data"



Semantic Web Search



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

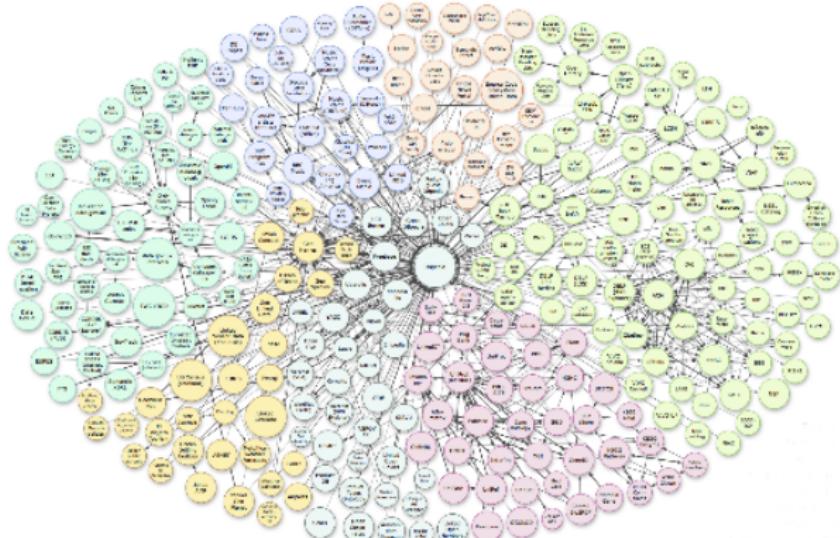
Spouse: Mirka Federer (m. 2009)

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



Semantic Web Search

Google

 $\exists X \text{ winnerOf}(X, \text{AustralianOpen2018})$ 

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

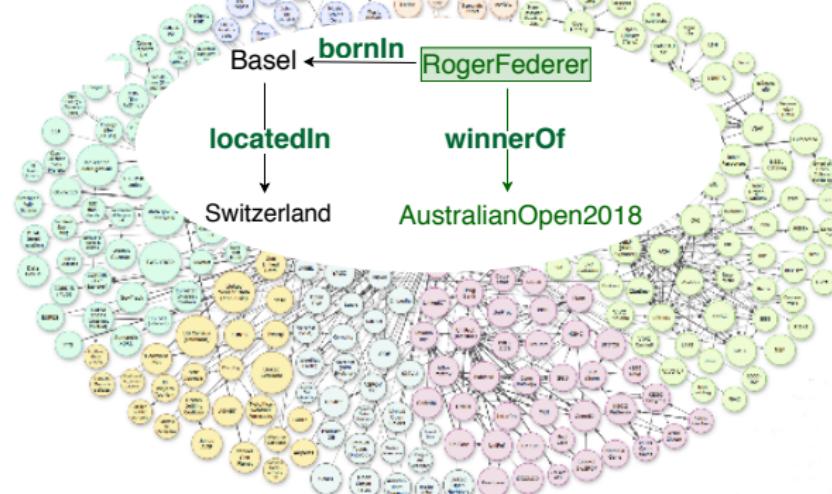
Weight: 85 kg

Spouse: Mirka Federer (m. 2009)

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

Semantic Web Search

Google

 $\exists X \text{ winnerOf}(X, \text{AustralianOpen2018})$ **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

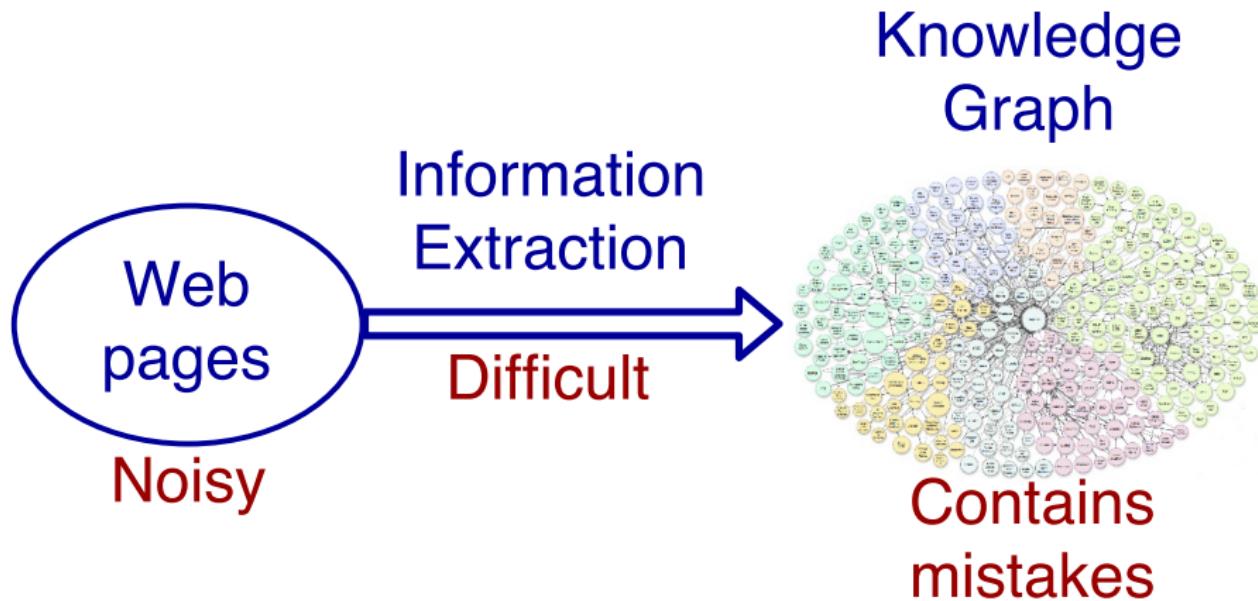
Height: 1.85 m

Weight: 85 kg

Spouse: Mirka Federer (m. 2009)

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

Problem: Inconsistency



Problem: Incompleteness

Google KG **misses** Roger's living place, but contains his wife's Mirka's..

living place of Roger Federer

living place of Mirka Federer

All Images News Videos Shopping More Settings Tools

All Images News Shopping Videos More Settings Tools

About 2 690 000 results (0,55 seconds)

About 1 910 000 results (0,92 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 ...

Mirka Federer / Residence



Bottmingen, Switzerland

What if we had rules?

$livesIn(Y, Z) \leftarrow marriedTo(X, Y), livesIn(X, Z)$, *Married people live together*



livesIn



What if we had rules?

*livesIn(Y, Z) ← 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



What if we had rules?

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

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

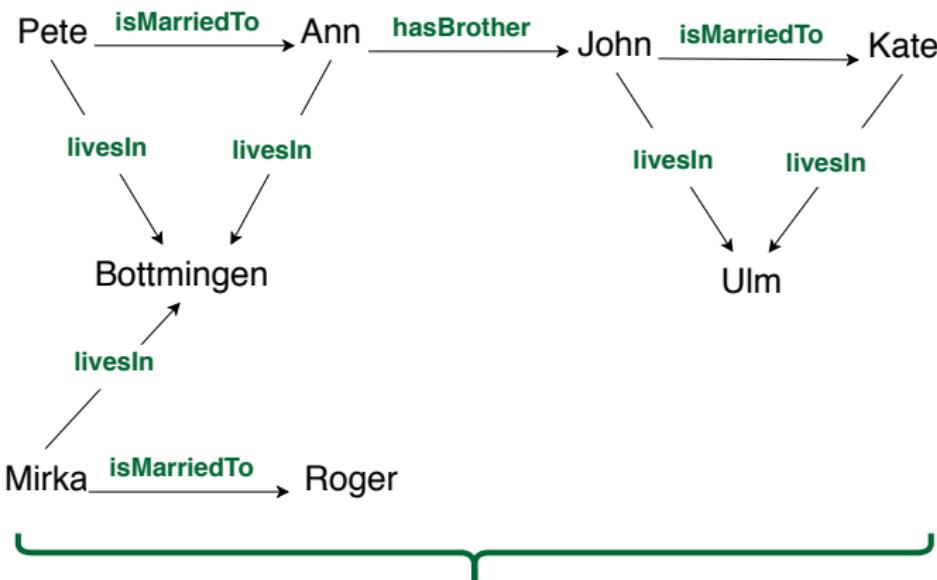


livesIn →



But where can one get such rules from?

Extracting Rules from Knowledge Graphs


$$\text{livesIn}(Y, Z) \leftarrow \text{isMarriedTo}(X, Y), \text{livesIn}(X, Z)$$

Motivation

Important problems of KGs:

- ① Inconsistency (covered in the morning!)
- ② Incompleteness (focus of this lecture)

Motivation

Important problems of KGs:

- ① Inconsistency (covered in the morning!)
- ② Incompleteness (focus of this lecture)

Inductive learning of common-sense rules for KG completion

Motivation

Preliminaries

Horn Rules

Nonmonotonic Rules

Rule Learning

Exception-awareness

Incompleteness

Rules from Hybrid Sources

Further Topics

Horn Rules

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

Informal semantics: If b_1, \dots, 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
 $\text{livesIn(roger, bottmingen)} \leftarrow \text{isMarried(mirka, roger)}, \text{livesIn(mirka, bottmingen)}$

Horn Rules

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

Informal semantics: If b_1, \dots, 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) ← isMarried(X, Y), livesIn(X, Z)
```

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 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

$$P = \{isMarriedTo(mirka, roger), livesIn(mirka, bottmingen), \\ livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), livesIn(X, Y)\}$$
$$HU(P) = \{mirka, roger, bottmingen\}$$
$$HB(P) = \{isMarriedTo(mirka, mirka), isMarriedTo(mirka, roger), \dots \\ livesIn(mirka, bottmingen), livesIn(roger, bottmingen), \dots\}$$
$$I_1 = \emptyset, I_2 = \{isMarriedTo(mirka, roger), livesIn(bottmingen, bottmingen)\}, \dots$$

Grounding Example

Program P :

```
p(X, Y, Z):-p(X, Y, Z'), h(X, Y), t(Z, Z', r).  
h(X, Z'):-p(X, Y, Z'), h(X, Y), t(Z, Z', r).  
p(0, 0, b).      h(0, 0).      t(a, b, r).
```

Grounding Example

Program P :

$$p(X, Y, Z) :- p(X, Y, Z'), h(X, Y), t(Z, Z', r).$$
$$h(X, Z') :- p(X, Y, Z'), h(X, Y), t(Z, Z', r).$$
$$p(0, 0, b). \quad h(0, 0). \quad t(a, b, r).$$

- The ground instances of the first rule are

$$p(0, 0, 0) :- p(0, 0, 0), h(0, 0), t(0, 0, r). \quad X = Y = Z = Z' = 0$$
$$\dots$$
$$p(0, r, 0) :- p(0, r, 0), h(0, r), t(0, 0, r). \quad X = Z = Z' = 0, Y = r$$
$$\dots$$
$$p(r, r, r) :- p(r, r, r), h(r, r), t(r, r, r). \quad X = Y = Z = Z' = r$$

- The single ground instance of the last rule is

$$t(a, b, r)$$

Herbrand Models

Def.: Herbrand models

An interpretation I is a (Herbrand) model of

- a ground (variable-free) clause $C = a:-b_1, \dots, b_m$, symbolically $I \models C$, if either $\{b_1, \dots, b_m\} \not\subseteq I$ or $a \in I$;
- a clause C , symbolically $I \models C$, if $I \models C'$ for every $C' \in \text{grnd}(C)$;
- a program P , symbolically $I \models P$, if $I \models C$ for every clause C in P .

Herbrand Models

Def.: Herbrand models

An interpretation I is a (Herbrand) model of

- a ground (variable-free) clause $C = a:-b_1, \dots, b_m$, symbolically $I \models C$, if either $\{b_1, \dots, b_m\} \not\subseteq I$ or $a \in I$;
- a clause C , symbolically $I \models C$, if $I \models C'$ for every $C' \in \text{grnd}(C)$;
- a program P , symbolically $I \models P$, if $I \models C$ for every clause C in P .

Proposition

For every positive logic program P , $\text{HB}(P)$ is a model of P .

Example

Reconsider program P :

$$\begin{aligned} p(X, Y, Z) &:- p(X, Y, Z'), h(X, Y), t(Z, Z', r). \\ h(X, Z') &:- p(X, Y, Z'), h(X, Y), t(Z, Z', r). \\ p(0, 0, b). \quad h(0, 0). \quad t(a, b, r). \end{aligned}$$

Which of the following interpretations are models of P ?

- $I_1 = \emptyset$
- $I_2 = HB(P)$
- $I_3 = \{h(0, 0), t(a, b, r), p(0, 0, b)\}$

Example

Reconsider program P :

$$\begin{aligned} p(X, Y, Z) &:- p(X, Y, Z'), h(X, Y), t(Z, Z', r). \\ h(X, Z') &:- p(X, Y, Z'), h(X, Y), t(Z, Z', r). \\ p(0, 0, b). \quad h(0, 0). \quad t(a, b, r). \end{aligned}$$

Which of the following interpretations are models of P ?

- $I_1 = \emptyset$ **no**
- $I_2 = HB(P)$
- $I_3 = \{h(0, 0), t(a, b, r), p(0, 0, b)\}$

Example

Reconsider program P :

$$\begin{aligned} p(X, Y, Z) &:- p(X, Y, Z'), h(X, Y), t(Z, Z', r). \\ h(X, Z') &:- p(X, Y, Z'), h(X, Y), t(Z, Z', r). \\ p(0, 0, b). \quad h(0, 0). \quad t(a, b, r). \end{aligned}$$

Which of the following interpretations are models of P ?

- $I_1 = \emptyset$ **no**
- $I_2 = HB(P)$ **yes**
- $I_3 = \{h(0, 0), t(a, b, r), p(0, 0, b)\}$

Example

Reconsider program P :

$$\begin{aligned} p(X, Y, Z) &:- p(X, Y, Z'), h(X, Y), t(Z, Z', r). \\ h(X, Z') &:- p(X, Y, Z'), h(X, Y), t(Z, Z', r). \\ p(0, 0, b). \quad h(0, 0). \quad t(a, b, r). \end{aligned}$$

Which of the following interpretations are models of P ?

- $I_1 = \emptyset$ **no**
- $I_2 = HB(P)$ **yes**
- $I_3 = \{h(0, 0), t(a, b, r), p(0, 0, b)\}$ **no**

Minimal Model Semantics

- A logic program has multiple models in general.
- Select one of these models as the canonical model.
- Commonly accepted: truth of an atom in model I should be “founded” by clauses.

Given:

$$P_1 = \{a \leftarrow b. \quad b \leftarrow c. \quad c\},$$

truth of a in the model $I = \{a, b, c\}$ is “founded”.

Given:

$$P_2 = \{a \leftarrow b. \quad b \leftarrow a. \quad c\},$$

truth of a in the model $I = \{a, b, c\}$ is not founded.

Minimal Model Semantics (cont'd)

Semantics follows Occam's razor principle: prefer models with true-part as small as possible.

Def: Minimal models

A model I of P is **minimal**, if there exists no model J of P such that $J \subset I$.

This is a consequence of the following property:

Proposition (Intersection closure)

If I and J are models of a positive program P , then $I \cap J$ is also a model of P .

Example

- For $P_1 = \{ a \leftarrow b. \quad b \leftarrow c. \quad c \}$, we have $LM(P_1) = \{a, b, c\}$.
- For $P_2 = \{ a \leftarrow b. \quad b \leftarrow a. \quad c \}$, we have $LM(P_2) = \{c\}$.
- For P from above,

$$p(X, Y, Z) \leftarrow p(X, Y, Z'), h(X, Y), t(Z, Z', r).$$

$$h(X, Z') \leftarrow p(X, Y, Z'), h(X, Y), t(Z, Z', r).$$

$$p(0, 0, b). \quad h(0, 0). \quad t(a, b, r).$$

we have

$$LM(P) = \{h(0, 0), t(a, b, r), p(0, 0, b), p(0, 0, a), h(0, b)\}.$$

Nonmonotonic Rules

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

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

Example

```
% Two married live together unless one is a researcher  
livesIn(Y, Z) ← isMarried(X, Y), livesIn(X, Z), not researcher(Y)
```

```
% Constraint: ensure that none is a parent of himself  
 $\perp \leftarrow \text{parent}(X, Y), \text{parent}(Y, X)$ 
```

Nonmonotonic Rules

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

Informal semantics: If b_1, \dots, b_m are true and none of b_{m+1}, \dots, 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"
 $\text{walk} \leftarrow \text{at}(L), \text{crossing}(L), \text{not train_approaches}(L)$

% At a rail road crossing cross the road if no train approaches
 $\text{walk} \leftarrow \text{at}(L), \text{crossing}(L), \neg \text{train_approaches}(L)$

Answer Set Semantics

Answer set program (ASP) \mathcal{P} is a set of nonmonotonic rules

$$\mathcal{P} = \left\{ \begin{array}{l} (1) \text{ } \textit{livesIn}(alex, ulm); \text{ } (2) \text{ } \textit{isMarried}(alex, mat); \\ (3) \text{ } \textit{livesIn}(Y, Z) \leftarrow \textit{isMarried}(X, Y), \textit{livesIn}(X, Z), \\ \quad \quad \quad \textit{not researcher}(Y); \end{array} \right\}$$

Answer Set Semantics

Evaluation of ASP programs is model-based¹, it consists of 2 steps:

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

$$\mathcal{P} = \left\{ \begin{array}{l} (1) \textit{livesIn}(alex, ulm); (2) \textit{isMarried}(alex, mat); \\ (3) \textit{livesIn}(mat, ulm) \leftarrow \textit{isMarried}(alex, mat), \textit{livesIn}(alex, ulm), \\ \quad \quad \quad \textit{not researcher}(mat); \end{array} \right\}$$

$$I = \{\textit{livesIn}(alex, ulm), \textit{isMarried}(alex, mat), \textit{livesIn}(mat, ulm)\}$$

CWA: *researcher(mat)* can not be derived, thus it is false

¹ unlike in prolog, which is based on theorem proving

Answer Set Semantics

Evaluation of ASP programs is model-based, it consists of 2 steps:

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

$$\mathcal{P} = \left\{ \begin{array}{l} (1) \text{ } livesIn(\text{alex}, \text{ulm}); \text{ (2) } isMarried(\text{alex}, \text{mat}); \\ (3) \text{ } livesIn(\text{mat}, \text{ulm}) \leftarrow isMarried(\text{alex}, \text{mat}), livesIn(\text{alex}, \text{ulm}), \\ \quad \quad \quad \text{not researcher}(\text{mat}); \\ (4) \text{ } researcher(\text{mat}) \end{array} \right\}$$

$$I = \{ \text{ } livesIn(\text{alex}, \text{ulm}), isMarried(\text{alex}, \text{mat}), \underline{livesIn(\text{mat}, \text{ulm})}, \text{ } researcher(\text{mat}) \}$$

Nonmonotonicity: adding facts might lead to loss of consequences!

Motivation

Preliminaries

Rule Learning

ILP

KG Challenges

Association Rules

Relational Association Rules

Exception-awareness

Incompleteness

Rules from Hybrid Sources

Reasoning with Incomplete Information

Default Reasoning

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

Abduction

choose between several explanations that explain an observation

Induction

generalize a rule from a number of similar observations

Reasoning with Incomplete Information

Default Reasoning

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

Abduction

choose between several explanations that explain an observation

Induction

generalize a rule from a number of similar observations

History of Inductive Learning

- **Philosophy of Science:**
Carnap, Hume, Miller, Popper, Peirce, ...
- **AI & Machine Learning 60s-70s:**
Banerji, Plotkin, Vere, Michalski, ...
- **AI & Machine Learning 80s:**
Shapiro, Sammut, Muggleton, ...
- **Inductive Logic Programming 1990s:**
Muggleton, Quinlan, De Raedt, ...
- **Statistical Relational Learning 2000s:**
Getoor, Koller, Domingos, Sato, ...

Zoo of ILP Tasks

ILP tasks can be classified along several dimensions:

- type of the data source, e.g., positive/negative examples, interpretations...

Zoo of ILP Tasks

ILP tasks can be classified along several dimensions:

- type of the data source, e.g., positive/negative examples, interpretations...
- type of the output knowledge, e.g., rules over single/multiple relations...

Zoo of ILP Tasks

ILP tasks can be classified along several dimensions:

- type of the data source, e.g., positive/negative examples, interpretations...
- type of the output knowledge, e.g., rules over single/multiple relations...
- the way the data is given as input, e.g., all at once, incrementally...

Zoo of ILP Tasks

ILP tasks can be classified along several dimensions:

- type of the data source, e.g., positive/negative examples, interpretations...
- type of the output knowledge, e.g., rules over single/multiple relations...
- the way the data is given as input, e.g., all at once, incrementally...
- availability of an oracle, e.g., human in the loop...

Zoo of ILP Tasks

ILP tasks can be classified along several dimensions:

- type of the data source, e.g., positive/negative examples, interpretations...
- type of the output knowledge, e.g., rules over single/multiple relations...
- the way the data is given as input, e.g., all at once, incrementally...
- availability of an oracle, e.g., human in the loop...
- quality of the data source, e.g., clean vs dirty...

Zoo of ILP Tasks

ILP tasks can be classified along several dimensions:

- type of the data source, e.g., positive/negative examples, interpretations...
- type of the output knowledge, e.g., rules over single/multiple relations...
- the way the data is given as input, e.g., all at once, incrementally...
- availability of an oracle, e.g., human in the loop...
- quality of the data source, e.g., clean vs dirty...
- data (in)completeness, e.g., OWA vs CWA...

Zoo of ILP Tasks

ILP tasks can be classified along several dimensions:

- type of the data source, e.g., positive/negative examples, interpretations...
- type of the output knowledge, e.g., rules over single/multiple relations...
- the way the data is given as input, e.g., all at once, incrementally...
- availability of an oracle, e.g., human in the loop...
- quality of the data source, e.g., clean vs dirty...
- data (in)completeness, e.g., OWA vs CWA...
- background knowledge, e.g., DL ontology, datalog rules, hybrid theories

Inductive Learning from Examples

Inductive Learning from Examples [?]

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)
- Syntactic restrictions on the definition of p

Inductive Learning from Examples

Inductive Learning from Examples [?]

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)
- Syntactic restrictions on the definition of p

Find:

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

Example

Given:

- $T = \{parentOf(john, mary), male(john), parentOf(david, steeve), male(david), parentOf(kathy, ellen), female(kathy)\}$
- $E^+ = \{fatherOf(john, mary), fatherOf(david, steve)\}$
- $E^- = \{fatherOf(kathy, ellen), fatherOf(john, steve)\}$
- Target language: Horn rules

Example

Given:

- $T = \{parentOf(john, mary), male(john), parentOf(david, steeve), male(david), parentOf(kathy, ellen), female(kathy)\}$
- $E^+ = \{fatherOf(john, mary), fatherOf(david, steve)\}$
- $E^- = \{fatherOf(kathy, ellen), fatherOf(john, steve)\}$
- Target language: Horn rules

Possible hypothesis:

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

Learning from Interpretations

Inductive Learning from Interpretations [?]

Given:

- I : interpretation, i.e., a set of facts over various relations
- T : background theory, i.e., a set of facts and possibly rules
- Syntactic restrictions on the form of rules to be induced

Learning from Interpretations

Inductive Learning from Interpretations [?]

Given:

- I : interpretation, i.e., a set of facts over various relations
- T : background theory, i.e., a set of facts and possibly rules
- Syntactic restrictions on the form of rules to be induced

Find:

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

Example

Inductive Learning from Interpretations [?]

Given:

- $I = \{isMarriedTo(mirka, roger), livesIn(mirka, b), livesIn(roger, b), bornIn(mirka, b)\}$
- $T = \{isMarriedTo(mirka, roger); bornIn(mirka, b); livesIn(X, Y) \leftarrow bornIn(X, Y)\}$
- Target rules: Horn rules

Example

Inductive Learning from Interpretations [?]

Given:

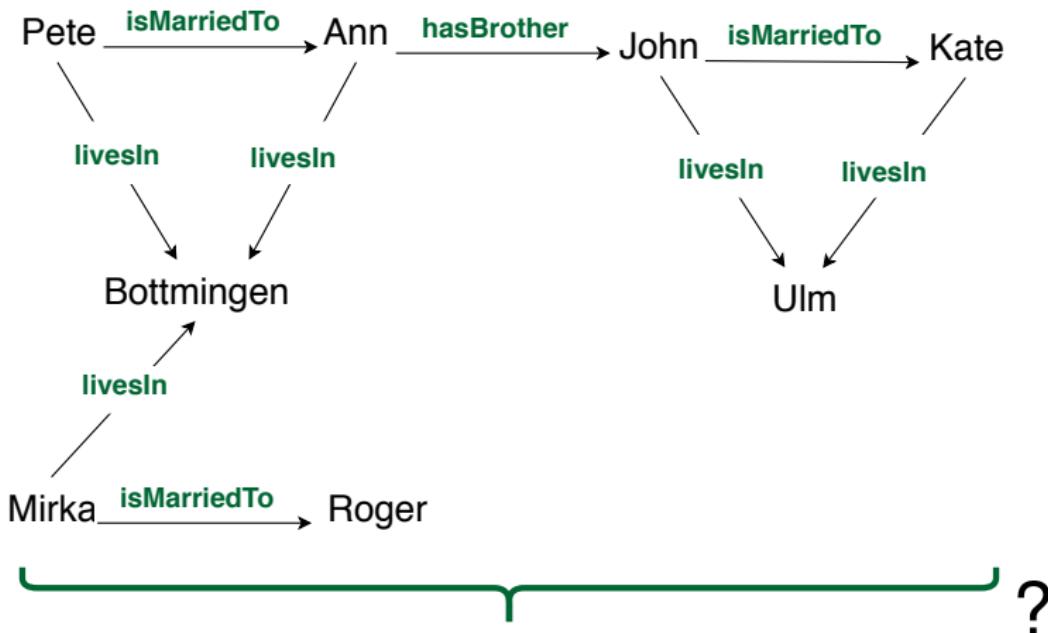
- $I = \{isMarriedTo(mirka, roger), livesIn(mirka, b), livesIn(roger, b), bornIn(mirka, b)\}$
- $T = \{isMarriedTo(mirka, roger); bornIn(mirka, b); livesIn(X, Y) \leftarrow bornIn(X, Y)\}$
- Target rules: Horn rules

Possible Hypothesis:

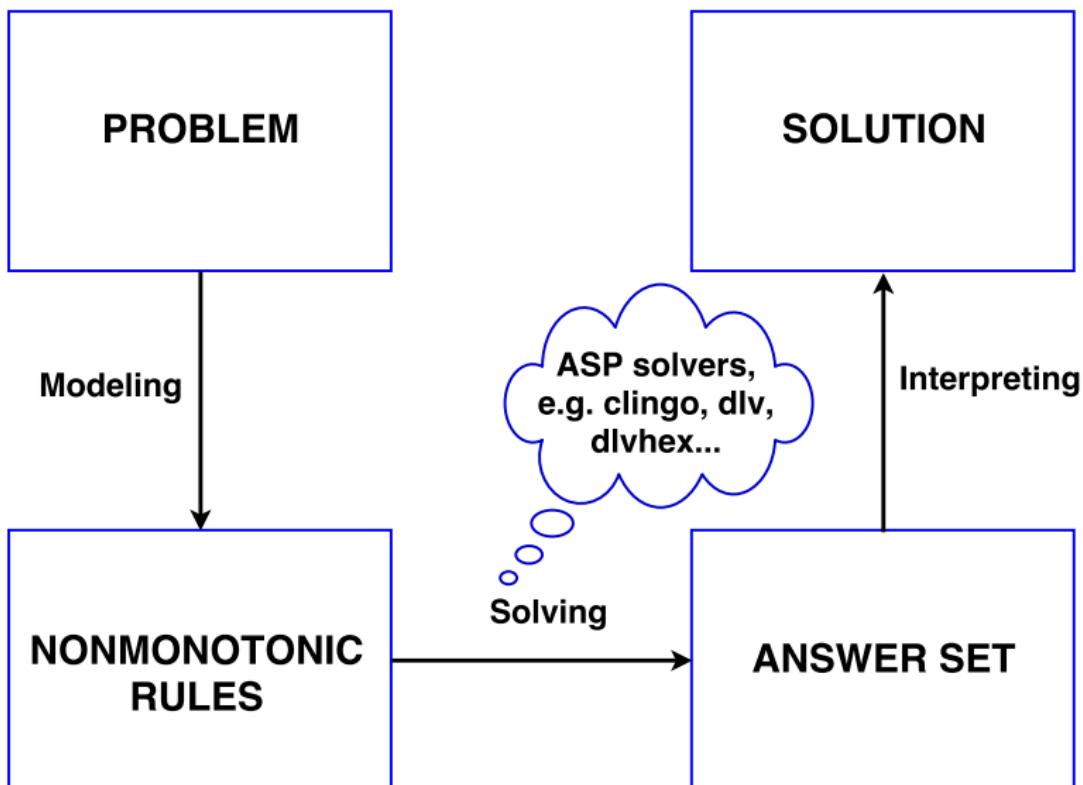
- $Hyp : livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), livesIn(Y, Z)$

Rule Induction from Knowledge Graphs

What is the most suitable setting for the task of inducing rules from KGs...

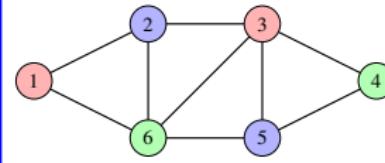
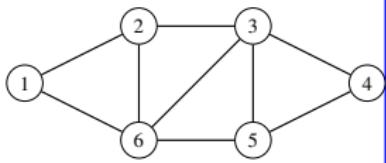


Declarative Programming



Inductive Learning from Interpretations

Graph 3-colorability



Modeling

```
node(1 .. 6); edge(1, 2); ...
col(V, red) ← not col(V, blue), not col(V, green), node(V);
col(V, green) ← not col(V, blue), not col(V, red), node(V);
col(V, blue) ← not col(V, green), not col(V, red), node(V);
⊥ ← col(V, C), col(V, C'), C ≠ C';
⊥ ← col(V, C), col(V', C), edge(V, V')
```

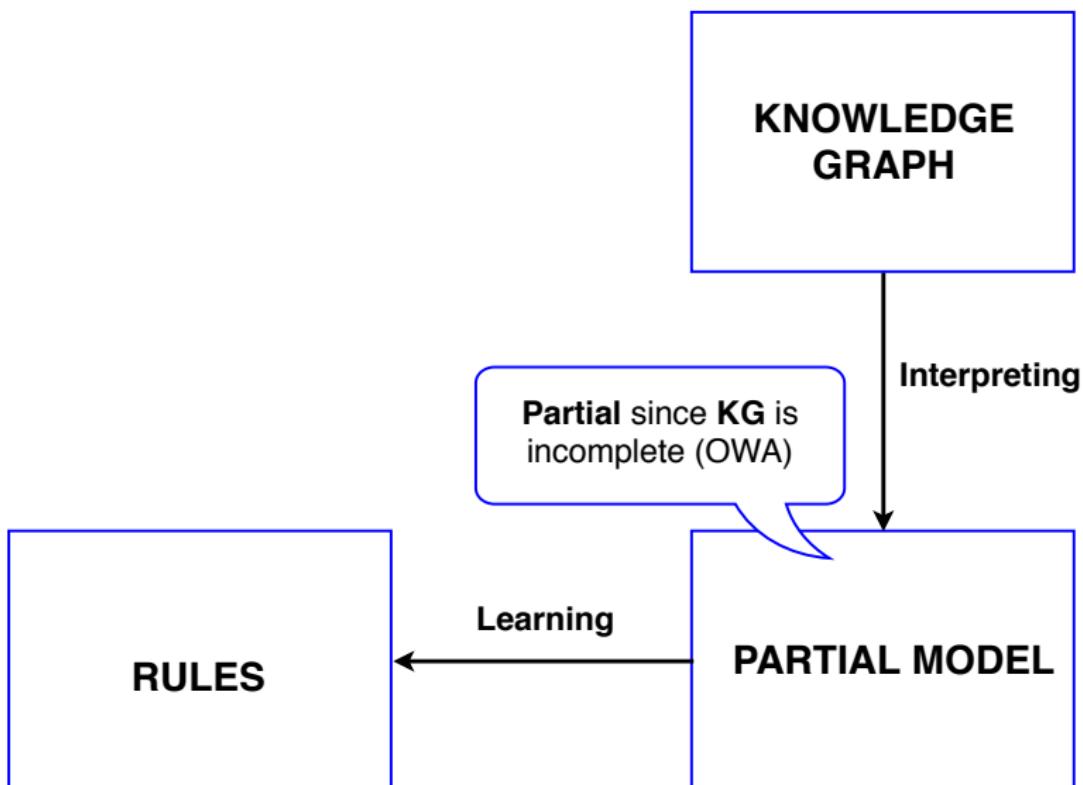
Interpreting

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)
```

Rule Induction



Challenges of Rule Induction from KGs

Open World Assumption: Negative examples are unavailable

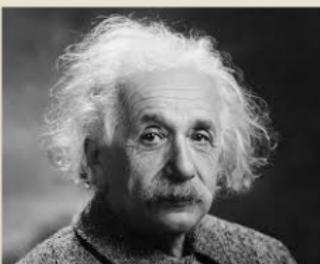
Challenges of Rule Induction from KGs

Open World Assumption: Negative examples are unavailable

Maybe Albert Einstein was a ballet dancer?

We dance for laughter,
we dance for tears,
we dance for madness,
we dance for fears,
we dance for hopes,
we dance for screams,
we are the dancers,
we create the dreams.

-Albert Einstein



Challenges of Rule Induction from KGs

Data bias: Only interesting facts are mentioned in text

Challenges of Rule Induction from KGs

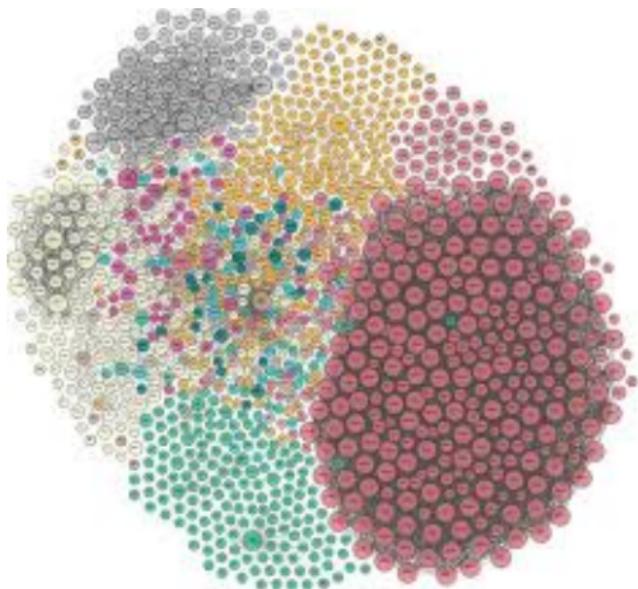
Data bias: Only interesting facts are mentioned in text

E.g., as wikipedia stores mostly famous people, one can induce wrong assumptions that all scientists are Nobel prize winners or all inhabitants of ancient Egypt are pharaohs..



Challenges of Rule Induction from KGs

Huge size: Modern KGs contain billions of facts



Challenges of Rule Induction from KGs

- **World knowledge is complex**, none of its “models” is perfect



No model is perfect

Exploratory Data Analysis

Question:

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

Answer:

Relational association rule mining!



Association Rules

- Which products are often bought together?
-

How to Apply Ideas to Relational Data?

Propositionalization

Relational Association Rule Mining

WARMER

AMIE

RDF2Rules

Ontology Path Finding

Motivation

Preliminaries

Rule Learning

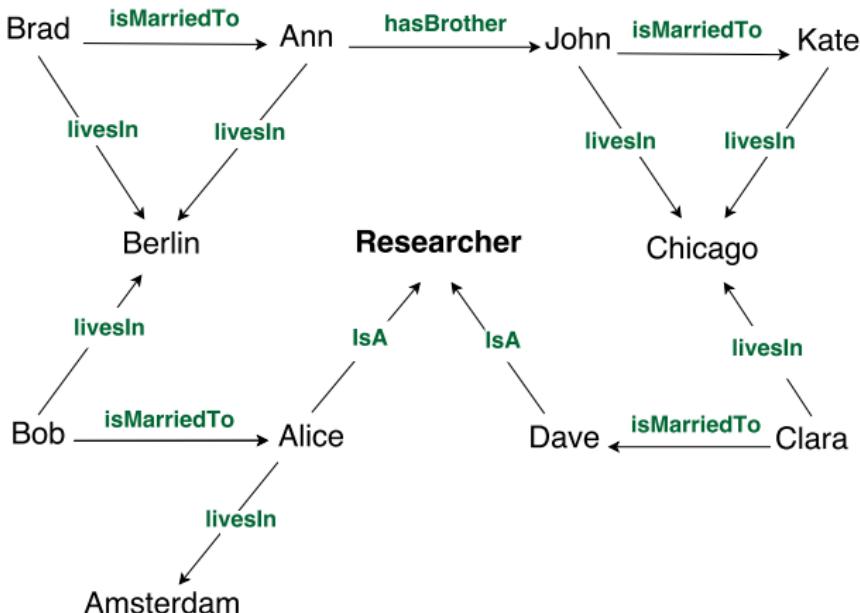
Exception-awareness

Incompleteness

Rules from Hybrid Sources

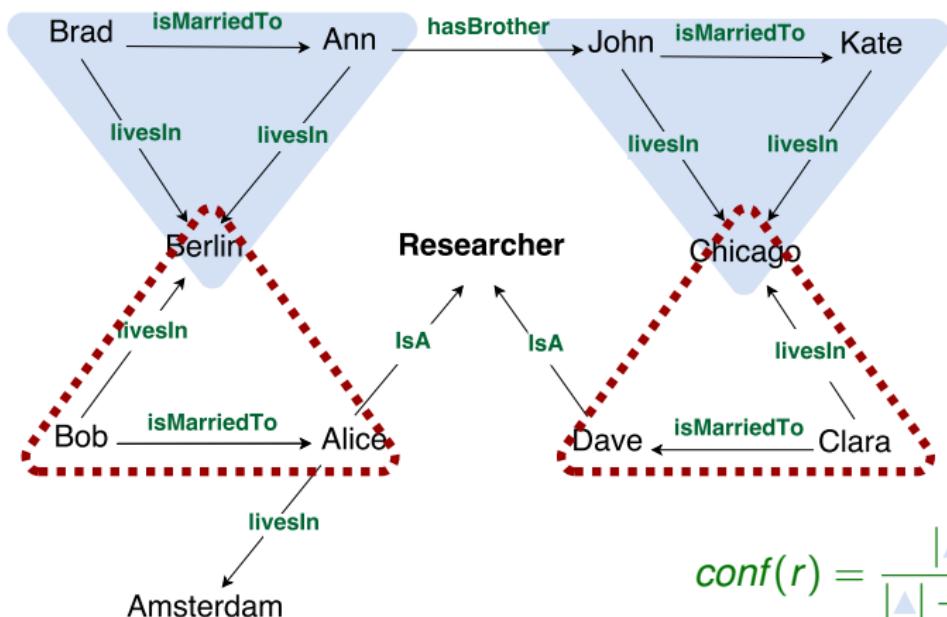
Further Topics

Horn Rule Mining



Horn Rule Mining

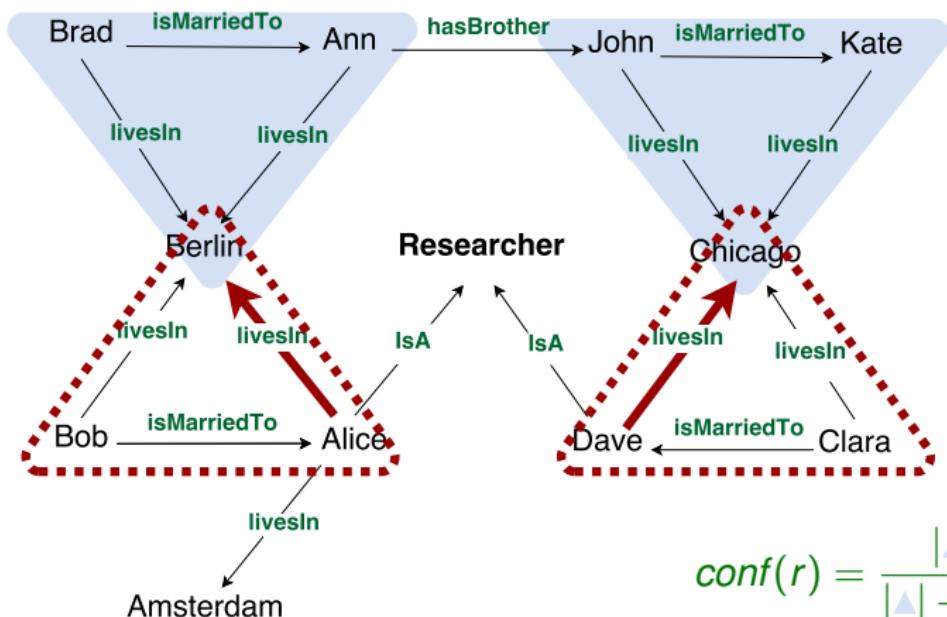
Horn rule mining for KG completion [Galárraga *et al.*, 2015]



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

Horn Rule Mining

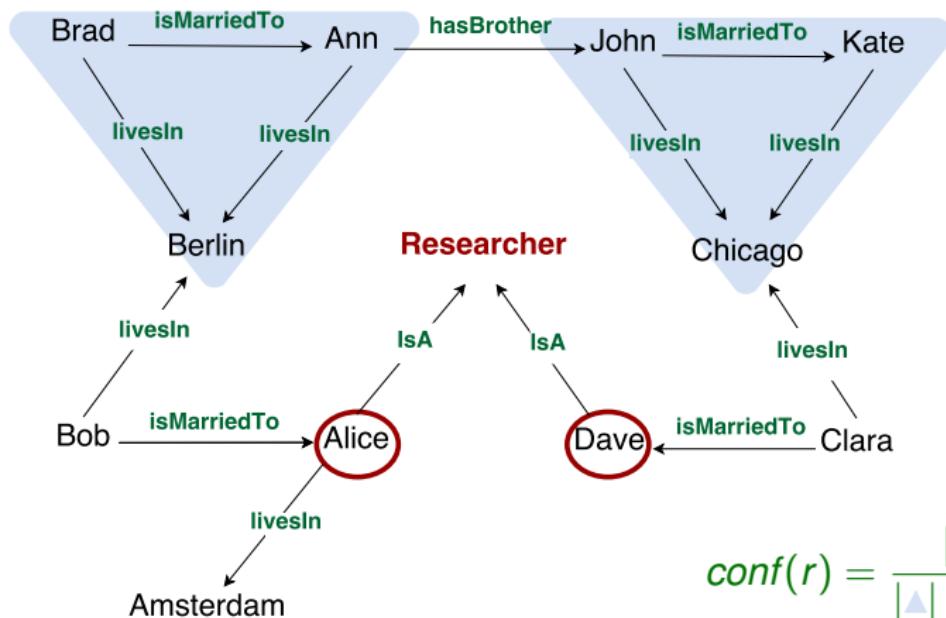
Horn rule mining for KG completion [Galárraga *et al.*, 2015]



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

Nonmonotonic Rule Mining

Nonmonotonic rule mining from KGs: OWA is a challenge!



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

Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG

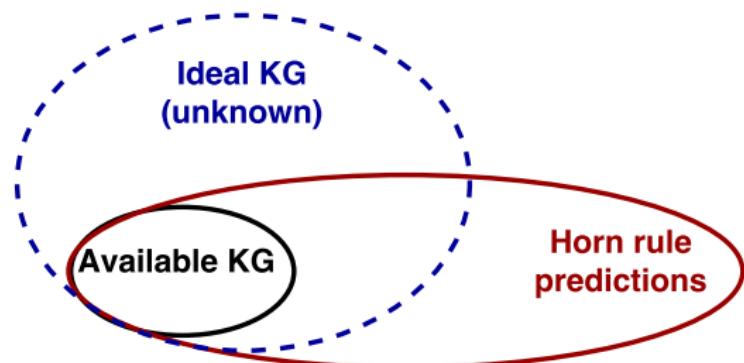


Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set

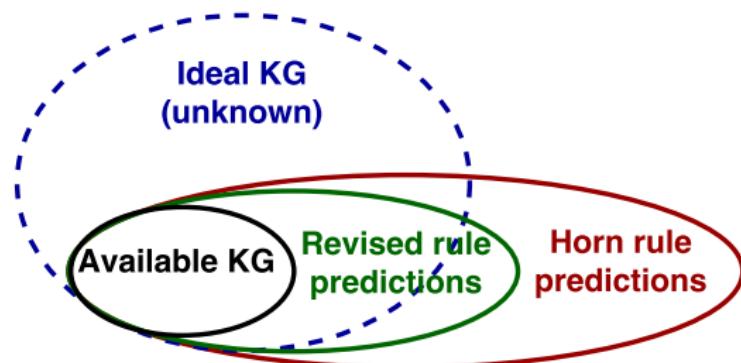


Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set



Find:

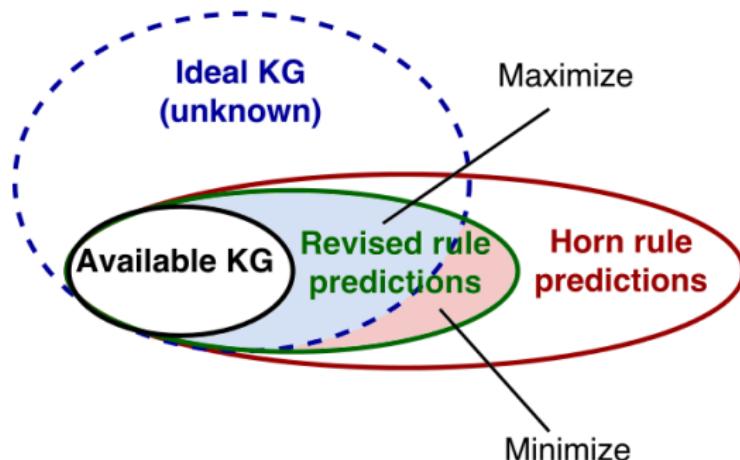
- Nonmonotonic revision of Horn rule set

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?

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

Avoid Data Overfitting

How to distinguish exceptions from noise?

$r1 : livesIn(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z), \text{not researcher}(X)$
 $\text{not_livesIn}(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z), researcher(X)$

Avoid Data Overfitting

How to distinguish exceptions from noise?

$r1 : \text{livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \text{not researcher}(X)$
 $\text{not_livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \text{researcher}(X)$

$r2 : \text{livesIn}(X, Z) \leftarrow \text{bornIn}(X, Z), \text{not moved}(X)$
 $\text{not_livesIn}(X, Z) \leftarrow \text{bornIn}(X, Z), \text{moved}(X)$

Avoid Data Overfitting

How to distinguish exceptions from noise?

$r1 : \text{livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \text{not researcher}(X)$
 $\text{not_livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \text{researcher}(X)$

$r2 : \text{livesIn}(X, Z) \leftarrow \text{bornIn}(X, Z), \text{not moved}(X)$
 $\text{not_livesIn}(X, Z) \leftarrow \text{bornIn}(X, Z), \text{moved}(X)$

$\{\text{livesIn}(c, d), \text{not_livesIn}(c, d)\}$ are conflicting predictions

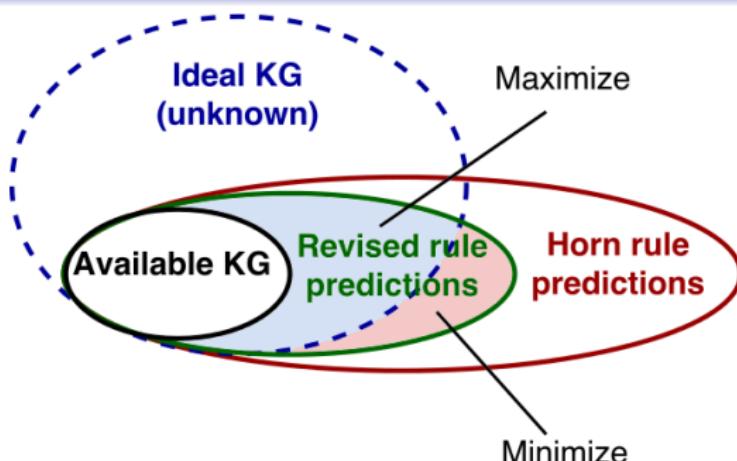
Intuition: Rules with good exceptions should make few conflicting predictions

Horn Theory Revision

Quality-based Horn Theory Revision

Given:

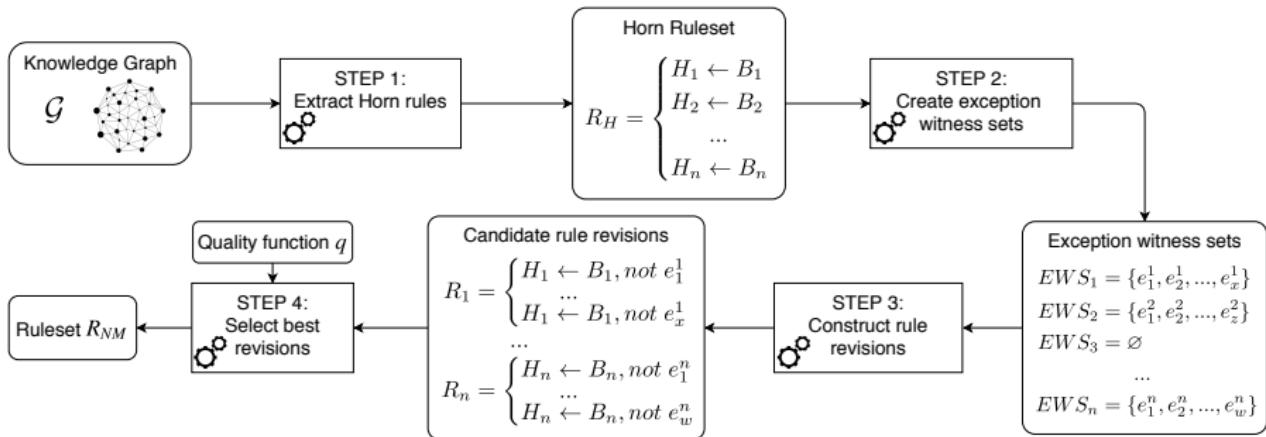
- Available KG
- Horn rule set



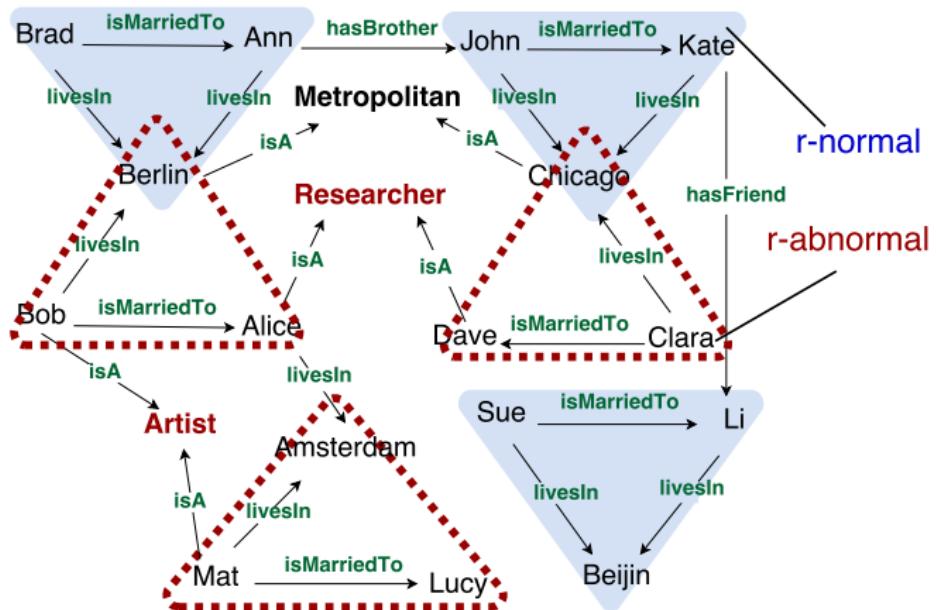
Find:

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

Approach Description



Exception Candidates



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

$\begin{cases} \text{not researcher}(X) \\ \text{not artist}(Y) \end{cases}$

Exception Ranking

rule1 $\{\underline{e_1}, e_2, e_3, \dots\}$

rule2 $\{e_1, \underline{e_2}, e_3, \dots\}$

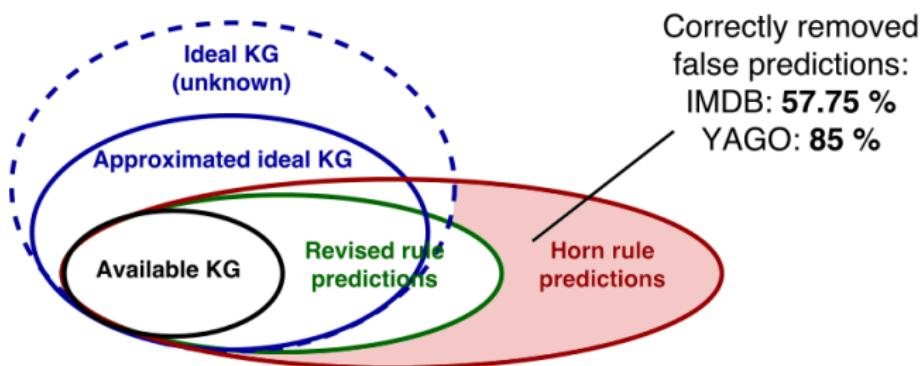
rule3 $\{\underline{e_1}, e_2, e_3, \dots\}$

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

Experimental Setup

- 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



Experimental Setup

- 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

$r_1 : \text{writtenBy}(X, Z) \leftarrow \text{hasPredecessor}(X, Y), \text{writtenBy}(Y, Z), \text{not american_film}(X)$

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

$r_2 : \text{actedIn}(X, Z) \leftarrow \text{isMarriedTo}(X, Y), \text{directed}(Y, Z), \text{not silent_film_actor}(X)$

Motivation

Preliminaries

Rule Learning

Exception-awareness

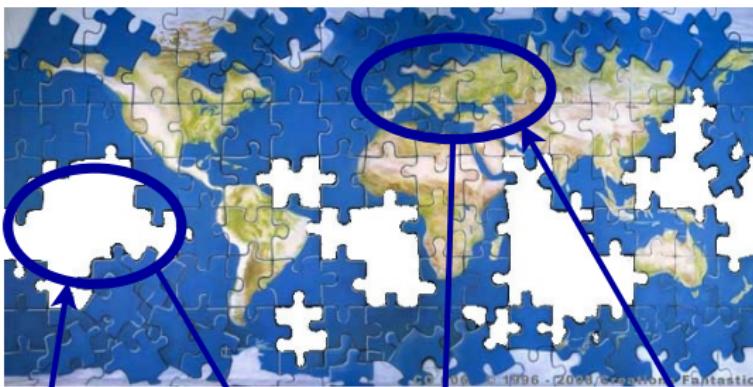
Incompleteness

Rules from Hybrid Sources

Further Topics

Completeness-aware Rule Mining

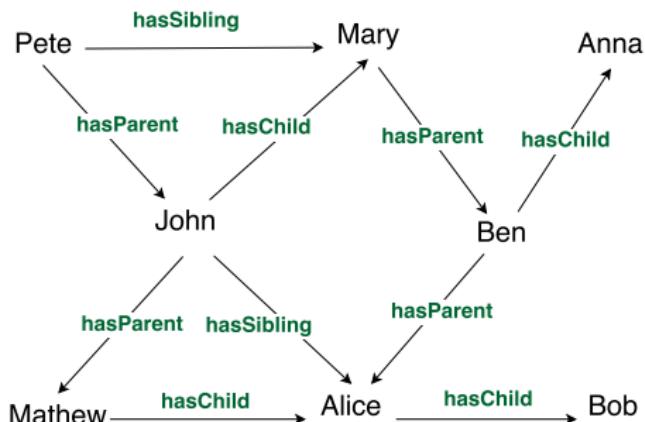
- Exploit cardinality meta-data [Mirza et al., 2016] in rule mining
John has 5 children, Mary is a citizen of 2 countries



build here! 5 missing do not build here!

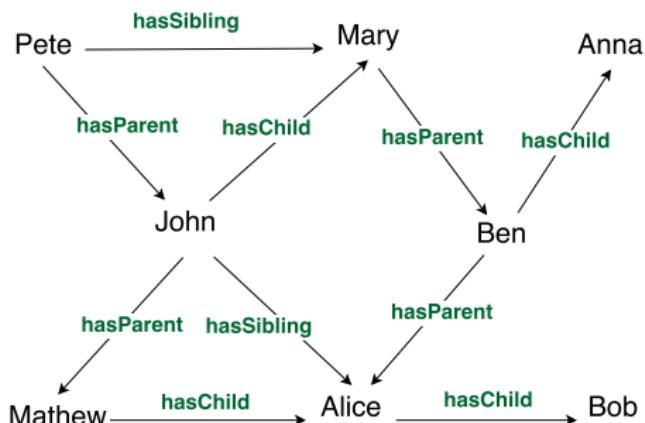
0 missing

Reasonable Rules



Reasonable Rules

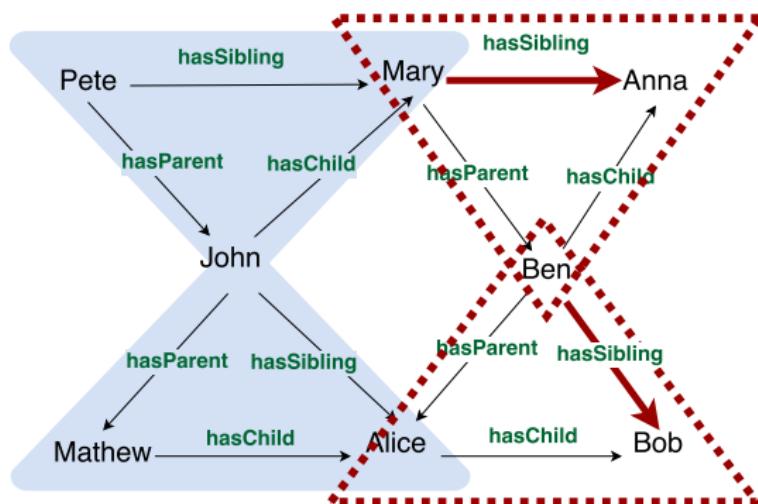
People with the same parents are likely siblings



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

Reasonable Rules

- ✓ *People with the same parents are likely siblings*

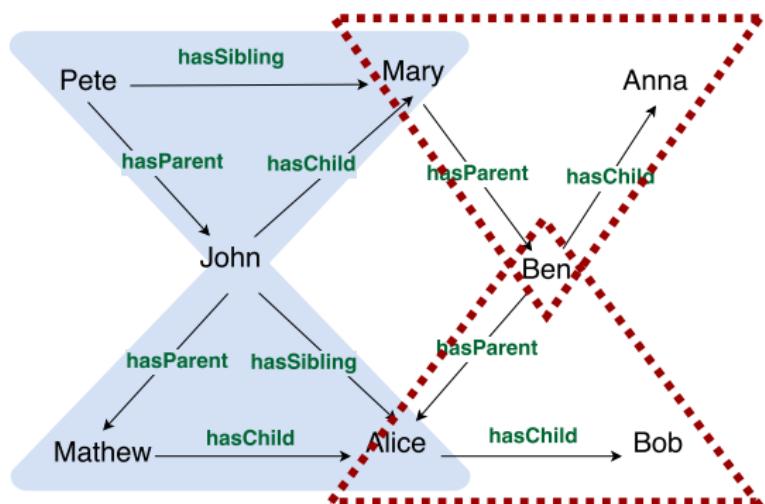


$r_1 : \text{hasSibling}(Z, Y) \leftarrow \text{hasChild}(X, Y), \text{hasParent}(Z, X)$

Reasonable Rules

✓ *People with the same parents are likely siblings*

Closed World Assumption (CWA): all children of Alice are known



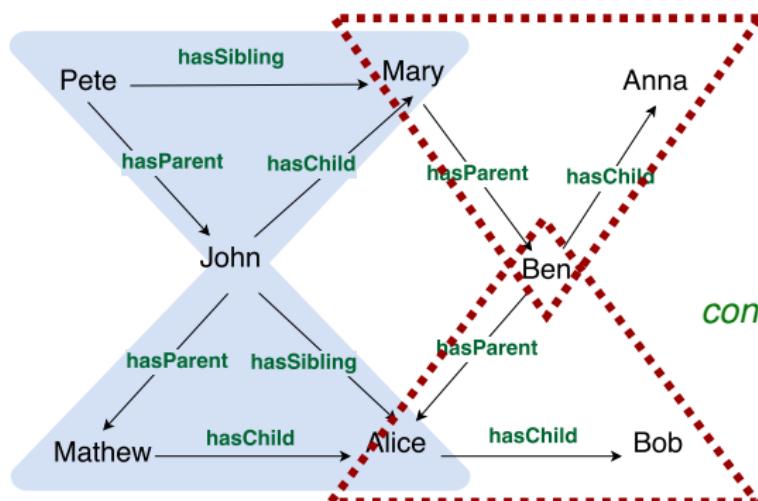
$$conf(r_1) = \frac{|\Delta|}{|\Delta| + |\triangle|} = \frac{2}{4}$$

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

Reasonable Rules

✓ *People with the same parents are likely siblings*

Partial Completeness A. (PCA): if a child of Alice is known, then all children are known [Galárraga et al., 2015]

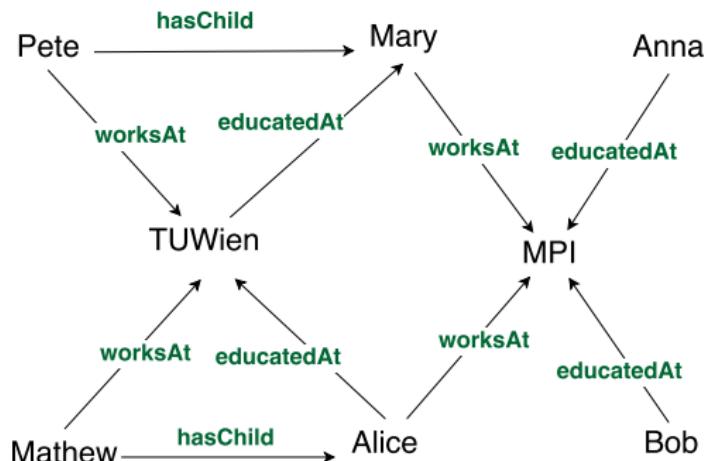


$$conf(r_1) = \frac{|\Delta|}{|\Delta| + |\triangle|} = \frac{2}{4}$$

$$conf_{pca}(r_1) = \frac{|\Delta|}{|\{\Delta | hasChild(Z, -) \in \mathcal{G}\}|} = \frac{2}{2}$$

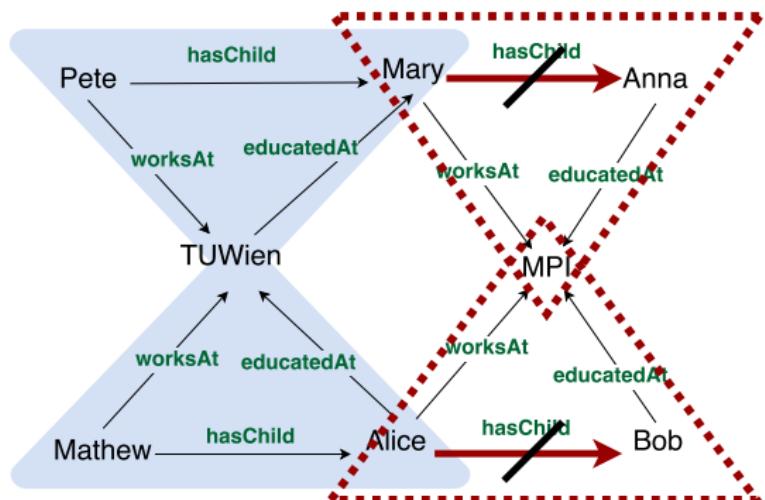
$r_1 : hasSibling(Z, Y) \leftarrow hasChild(X, Y), hasParent(Z, X)$

Erroneous Rules due to Data Bias



Erroneous Rules due to Data Bias

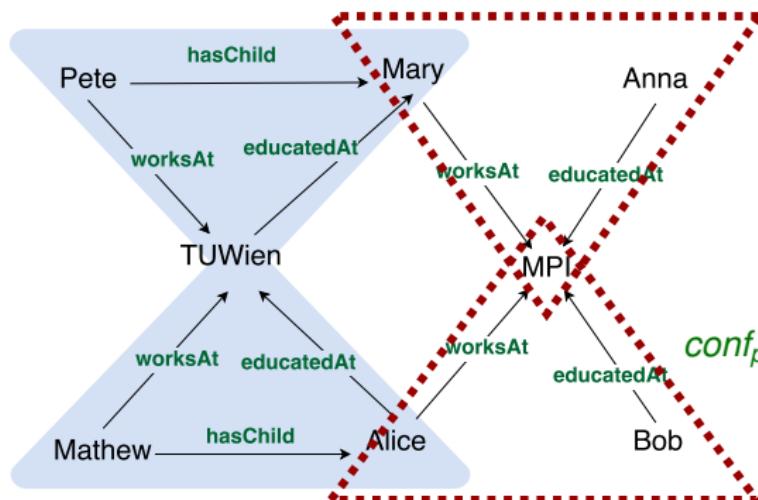
People working and studying at the same institute are likely relatives



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

Erroneous Rules due to Data Bias

✗ People working and studying at the same institute are likely relatives



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

$$conf_{pca}(r_2) = \frac{|\Delta|}{|\{\Delta | hasSibling(Z, -) \in \mathcal{G}\}|} = \frac{2}{2}$$

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

Exploiting Meta-data in Rule Learning

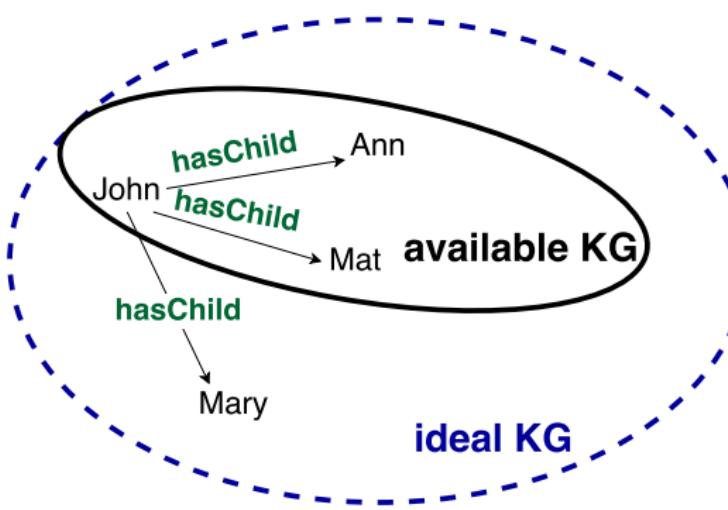
Goal: make use of cardinality constraints on edges of the KG to improve rule learning.



5 missing
0 missing
build here!
do not build here!

Cardinality Statements

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

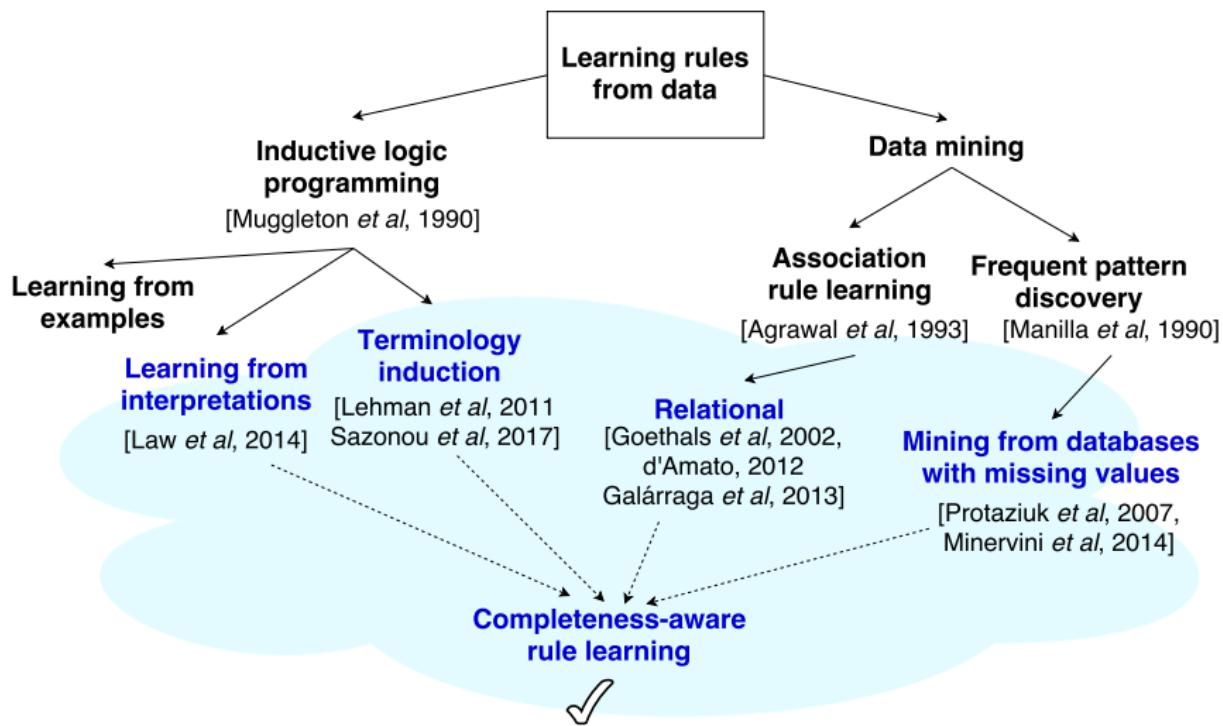


$\text{num}(\text{hasChild}, \text{john}) = 3$
 $\text{miss}(\text{hasChild}, \text{john}) = 1$
 $\text{incomplete}(\text{hasChild}, \text{john})$

Cardinality Constraints on Edges

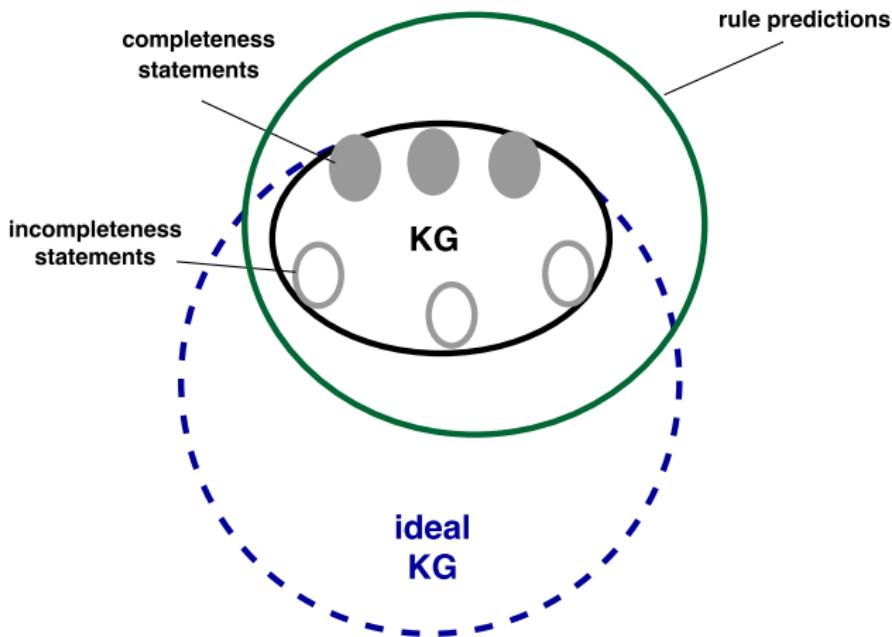
- Mining cardinality assertions from the Web [?]
 - “... *John has 2 children ...*”
- Estimating recall of KGs by crowd sourcing [?]
 - *20 % of Nobel laureates in physics are missing*
- Predicting completeness in KGs [?]
 - Add *complete(john, hasChild)* to KG and mine rules
 $complete(X, hasChild)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

Related Work



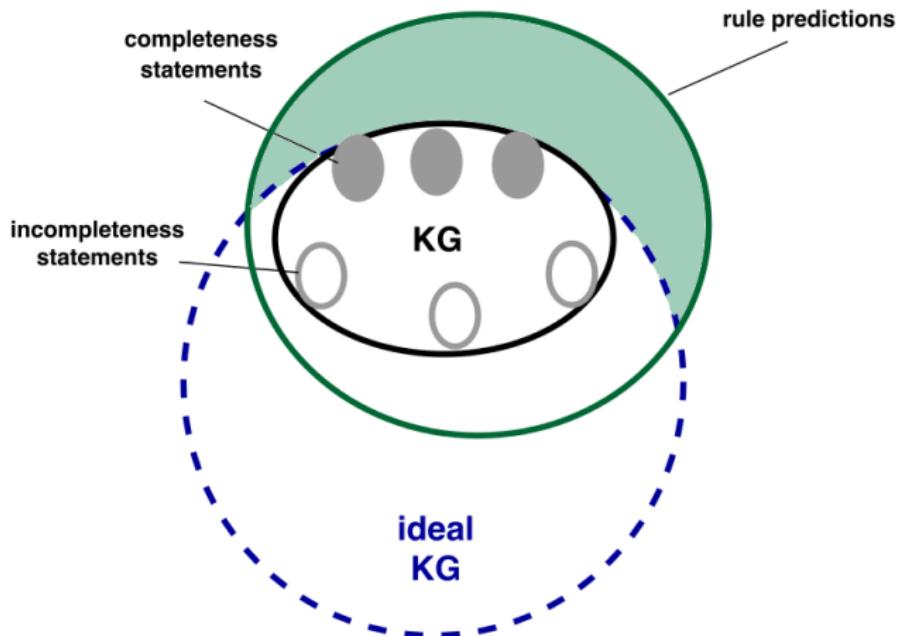
Prediction Post-processing

Remove predictions in complete KG parts [?],
i.e., constraints are set on the output not the input



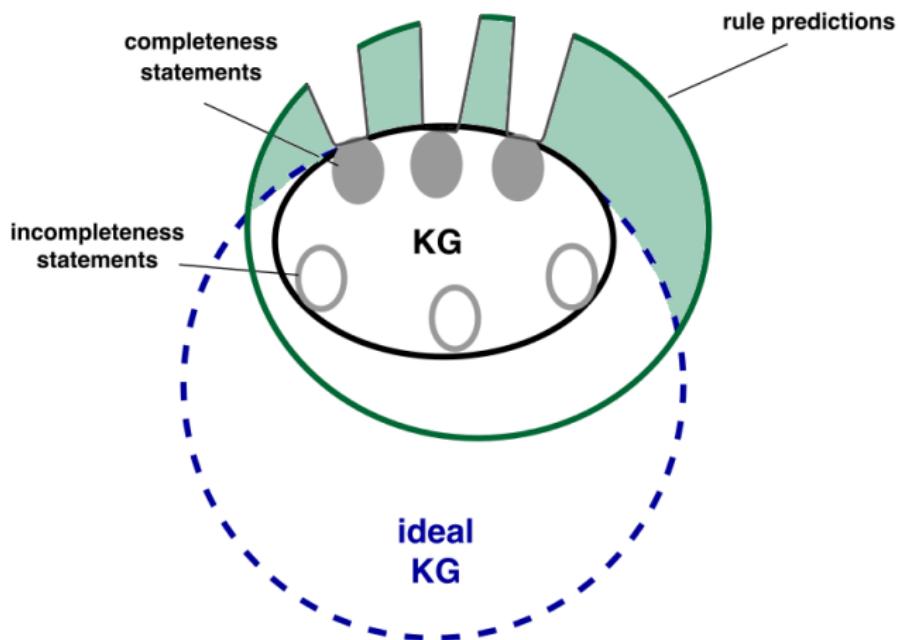
Prediction Post-processing

Remove predictions in complete KG parts [?],
i.e., constraints are set on the output not the input



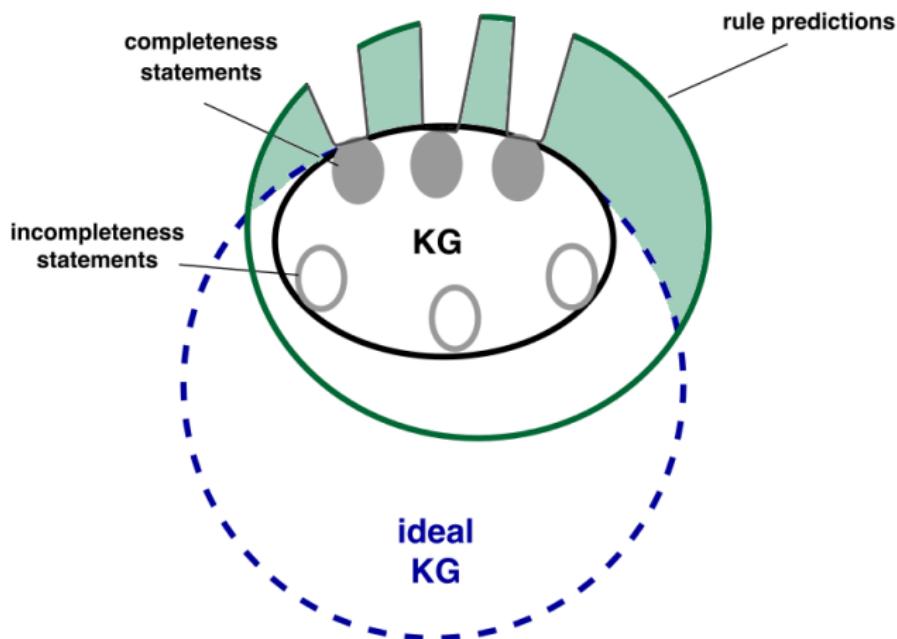
Prediction Post-processing

Remove predictions in complete KG parts [?],
i.e., constraints are set on the output not the input



Prediction Post-processing

Remove predictions in complete KG parts [?],
i.e., constraints are set on the output not the input

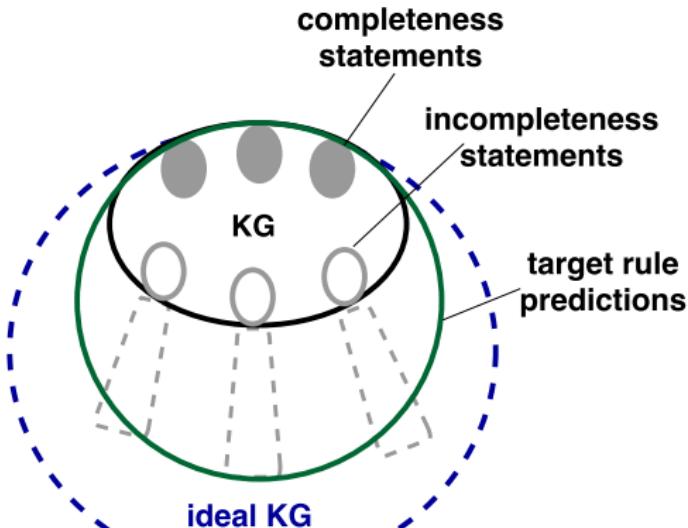


Rules might be still **erroneous**.. What about other incorrect predictions?

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

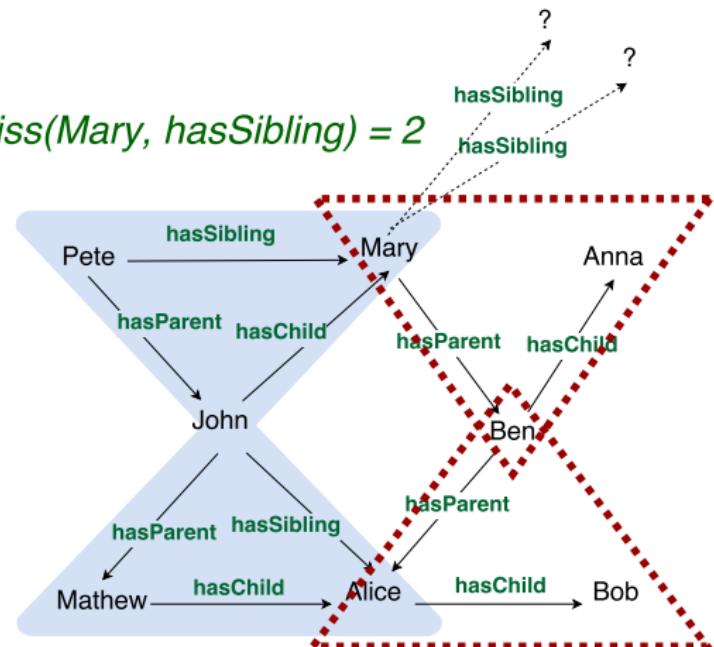
$npi(r)$: number of facts added to incomplete areas by r

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

Rule Predictions

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

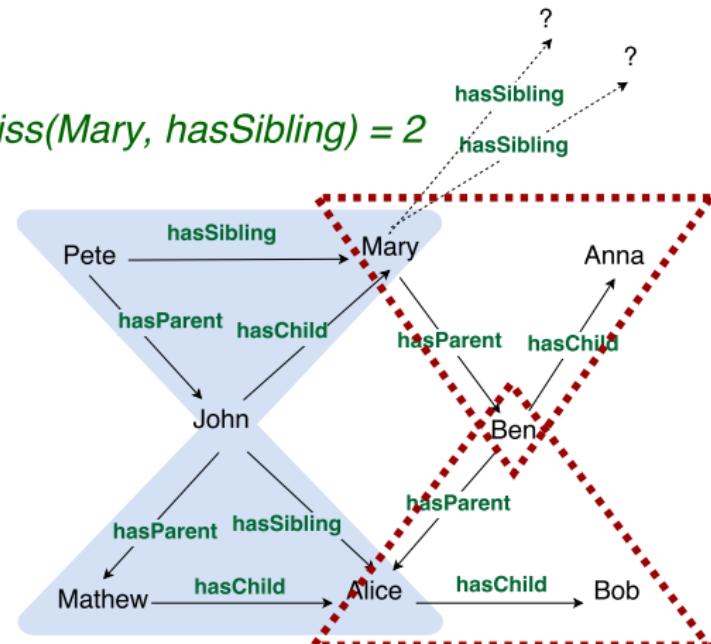
$npc(r)$: number of facts added to complete areas by r



Rule Predictions

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

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



$miss(Mary, hasSibling) = 2$

$$npi(r_1) = 1$$

$$npc(r_1) = 0$$

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

Completeness Confidence

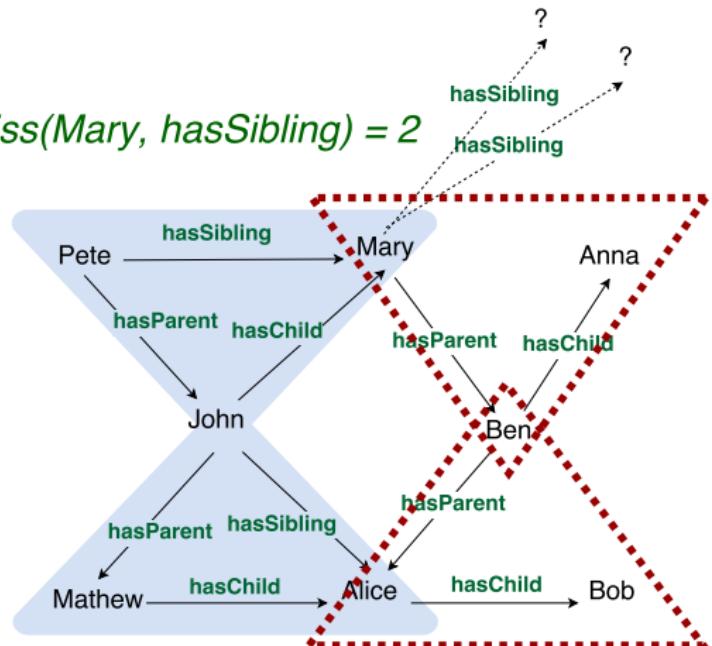
$conf_{comp}$: do not penalize rules that predict new facts in incomplete areas

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

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

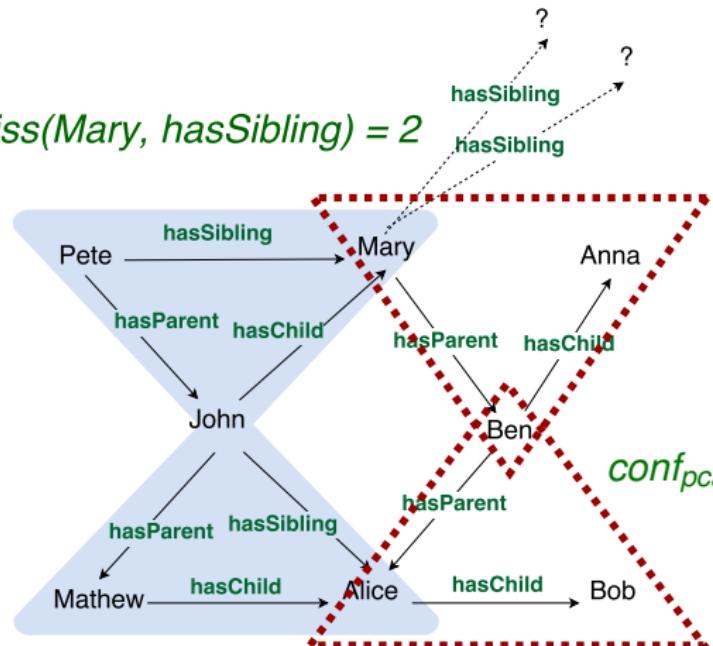
Completeness Confidence Example 1

$\text{miss}(\text{Mary}, \text{hasSibling}) = 2$



Completeness Confidence Example 1

$\text{miss}(\text{Mary}, \text{hasSibling}) = 2$



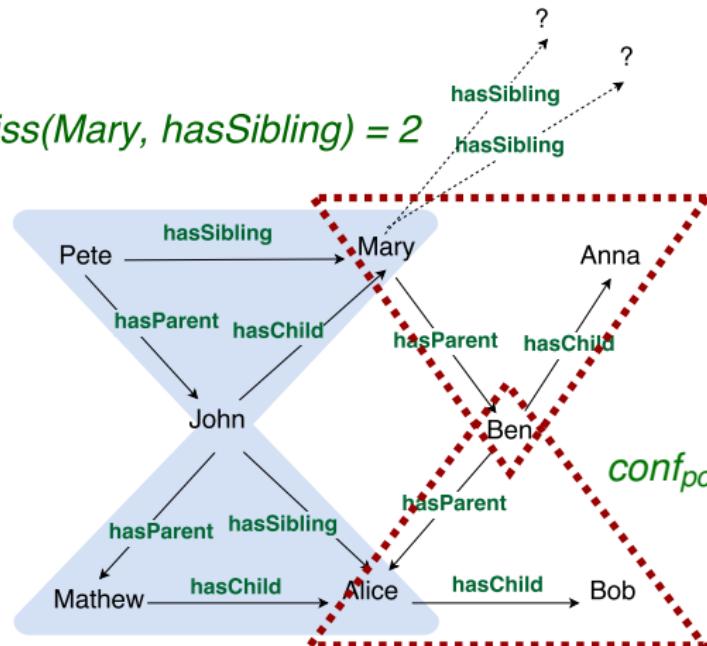
$$\text{conf}(r_1) = \frac{|\triangle|}{|\triangle| + |\triangle|} = \frac{2}{4}$$

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

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

Completeness Confidence Example 1

$\text{miss}(\text{Mary}, \text{hasSibling}) = 2$



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

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

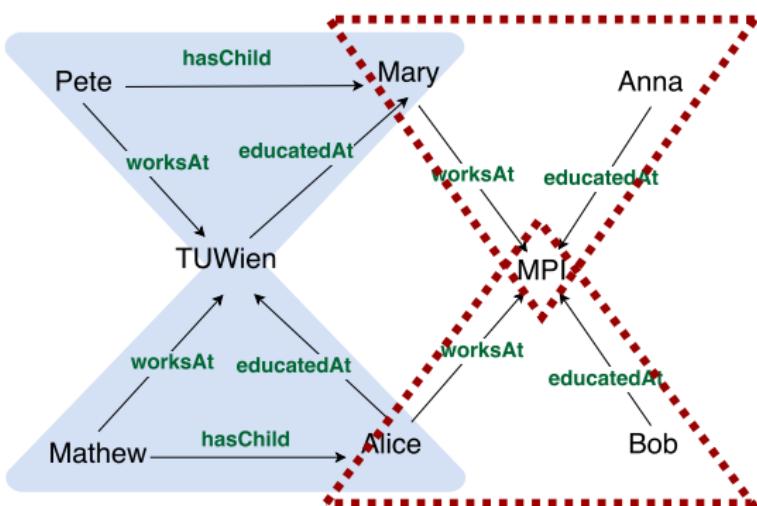
$$\text{npi}(r_1) = 1$$

$$\text{conf}_{\text{comp}}(r_1) = \frac{|\triangle|}{|\triangle| + |\triangle| - \text{npi}(r_1)} = \frac{2}{3}$$

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

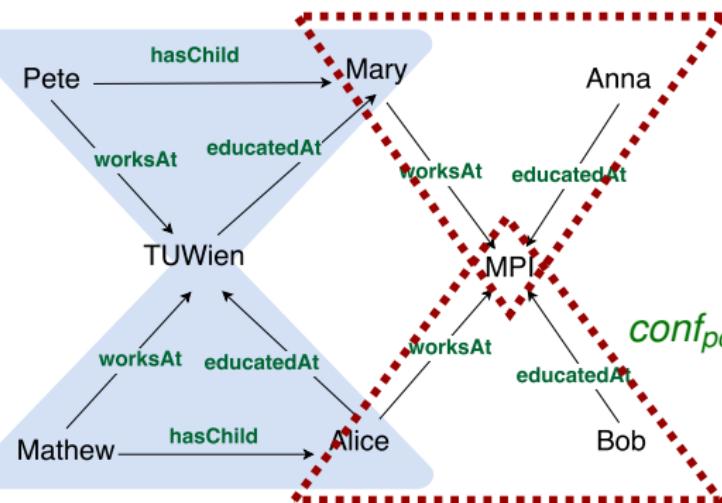
Completeness Confidence Example 2

$\text{miss}(\text{hasChild}, \text{Alice}) = 0$



Completeness Confidence Example 2

$\text{miss}(\text{hasChild}, \text{Alice}) = 0$



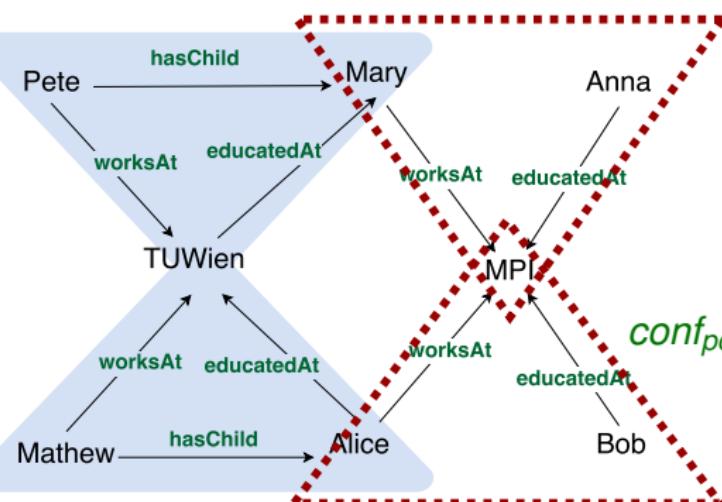
$$\text{conf}(r_2) = \frac{|\triangle|}{|\triangle| + |\triangle|} = \frac{2}{4}$$

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

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

Completeness Confidence Example 2

$$\text{miss}(\text{hasChild}, \text{Alice}) = 0$$



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

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

$$\text{npi}(r_2) = 0$$

$$\text{conf}_{\text{comp}}(r_2) = \frac{|\triangle|}{|\triangle| + |\triangle| - \text{npi}(r_2)} = \frac{2}{4}$$

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

Other Measures

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

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

$recall_{comp}$: 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

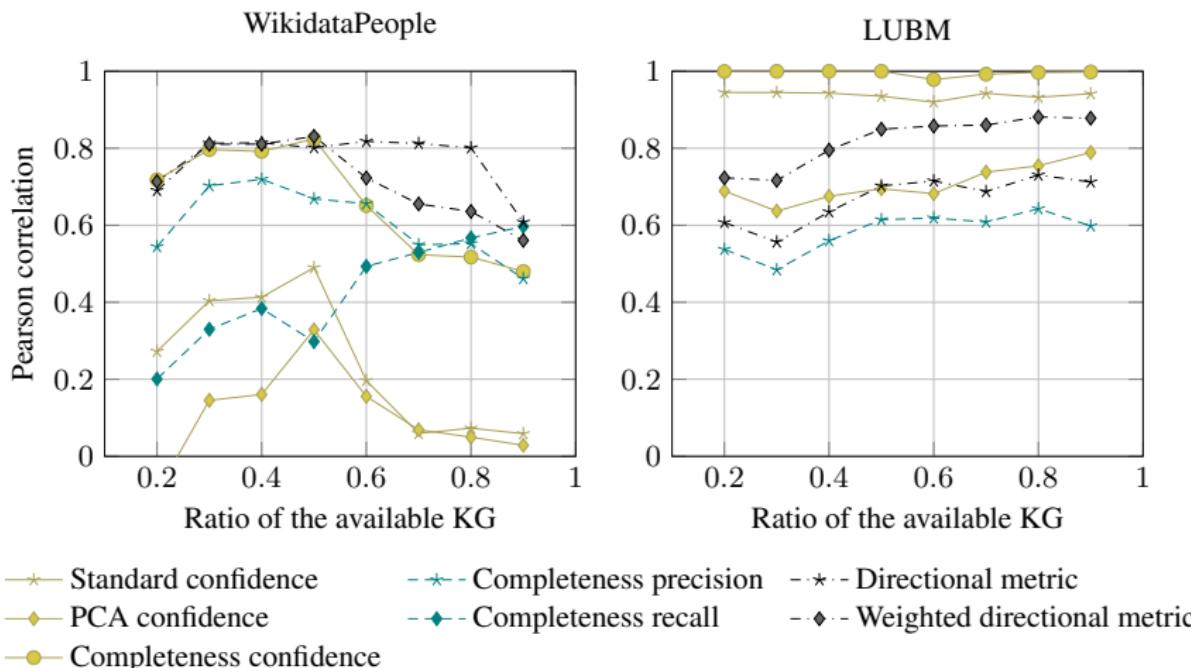
Creation of ideal KG:

- 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



Cardinality statements mining

- Introduce $p_{\geq k}(s)$ and $p_{\leq k}(s)$ for each $\text{num}(p, s)$
- Introduce $p_{\geq |\{o| (s, p, o) \in \mathcal{G}\}|}(s)$ for all p and s
- Use the background rules $p_{\geq k}(s) \leftarrow p_{\geq k+1}(s)$ and $p_{\leq k+1}(s) \leftarrow p_{\leq k}(s)$.
- Mine rules which head is a $p_{_}(_)$
- Complete the \mathcal{G} with a confidence threshold
- if $p_{\geq k}(s) \in \mathcal{G}_c$ and $p_{\leq k}(s) \in \mathcal{G}_c$ then $\text{num}(p, s) = k$

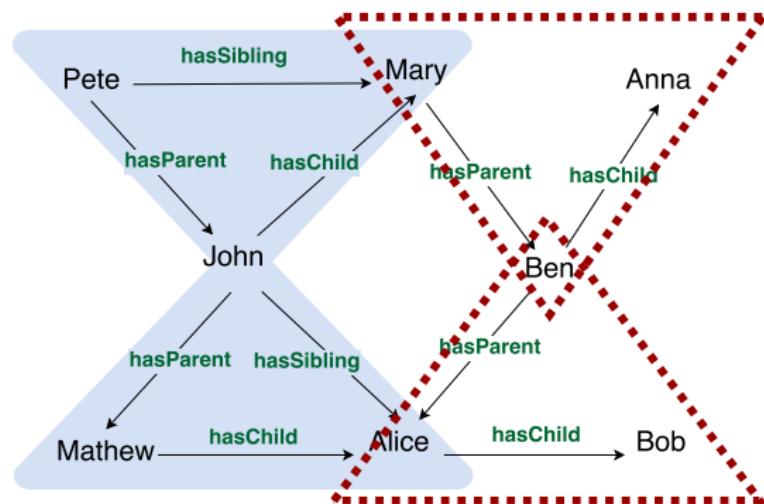
Completeness precision and recall

We define a precision and a recall:

$$\text{precision}_{\text{comp}}(r) = 1 - \frac{\text{npc}(r)}{\text{supp}(\mathbf{B})} \text{ (ratio of "complete" results)}$$

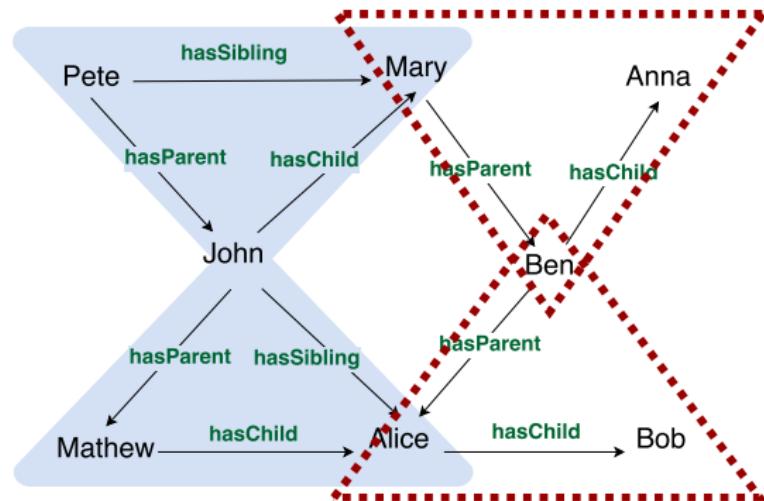
$$\text{recall}_{\text{comp}}(r) = \frac{\text{npi}(r)}{\sum_s \text{miss}(h,s)} \text{ (ratio of "incomplete" results filled)}$$

Example with precision and recall



$r : \text{hasSibling}(Z, Y) \leftarrow \text{hasChild}(X, Y), \text{hasParent}(Z, X)$
 $\text{miss}(\text{hasSibling}, \text{Mary}) = 2$

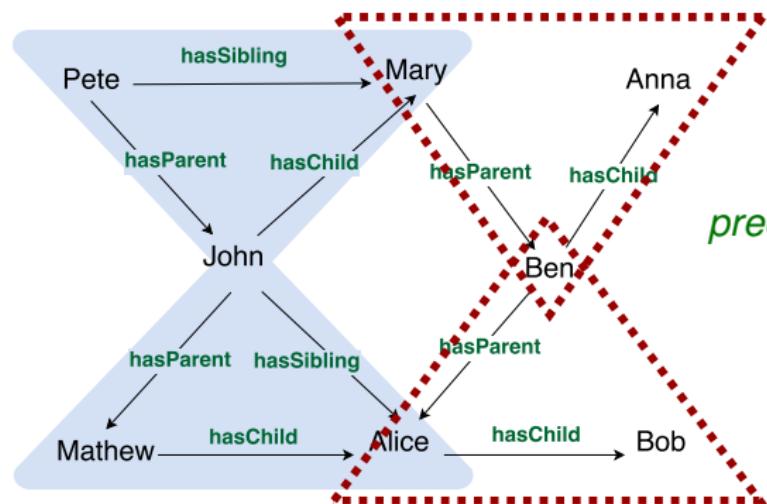
Example with precision and recall



$$npi(r) = 1 \quad npc(r) = 0$$

$r : hasSibling(Z, Y) \leftarrow hasChild(X, Y), hasParent(Z, X)$
 $miss(hasSibling, Mary) = 2$

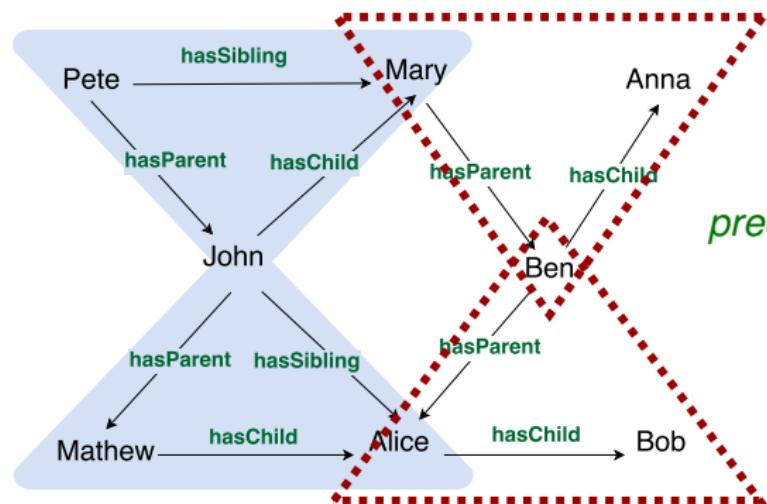
Example with precision and recall



$$\begin{aligned}
 npi(r) &= 1 & npc(r) &= 0 \\
 precision_{comp}(r) &= 1 - \frac{0}{4} = 1
 \end{aligned}$$

$r : hasSibling(Z, Y) \leftarrow hasChild(X, Y), hasParent(Z, X)$
 $miss(hasSibling, Mary) = 2$

Example with precision and recall



$$\begin{aligned}
 npi(r) &= 1 & npc(r) &= 0 \\
 precision_{comp}(r) &= 1 - \frac{0}{4} = 1 & recall_{comp}(r) &= \frac{1}{2}
 \end{aligned}$$

$r : hasSibling(Z, Y) \leftarrow hasChild(X, Y), hasParent(Z, X)$
 $miss(hasSibling, Mary) = 2$

Directional metric

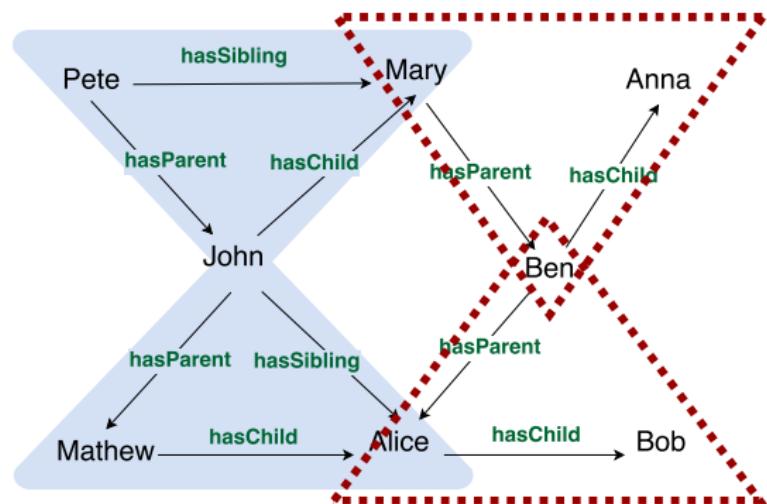
$$\text{dir_metric}(r) = \frac{npi(r) - npc(r)}{2 \cdot (npi(r) + npc(r))} + 0.5$$

Directional metric

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

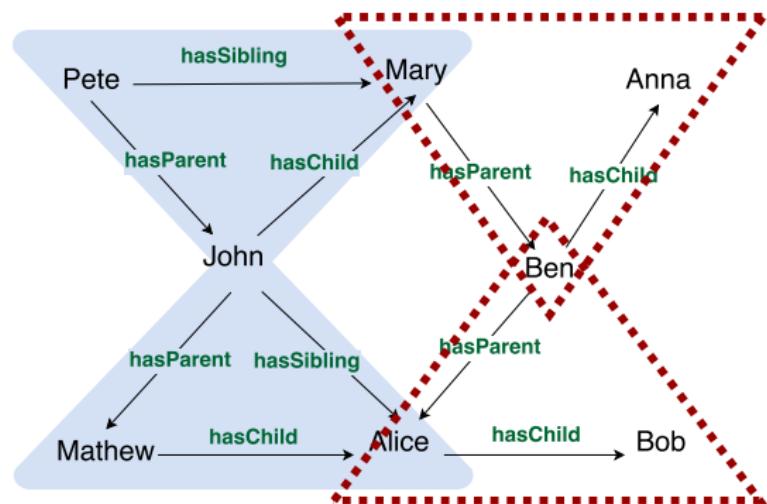
Weighted: $\text{wdm}(r) = \beta \cdot \text{conf}(r) + (1 - \beta) \cdot \text{dir_metric}(r)$

Example with directional metric



$r : \text{hasSibling}(Z, Y) \leftarrow \text{hasChild}(X, Y), \text{hasParent}(Z, X)$
 $\text{miss}(\text{hasSibling}, \text{Mary}) = 2 \beta = 0.5$

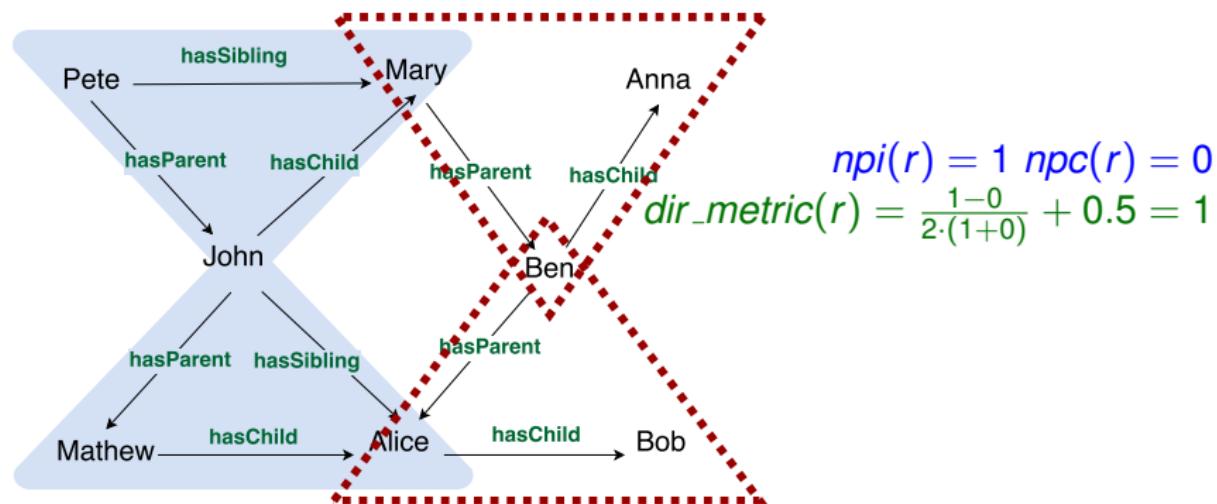
Example with directional metric



$$npi(r) = 1 \quad npc(r) = 0$$

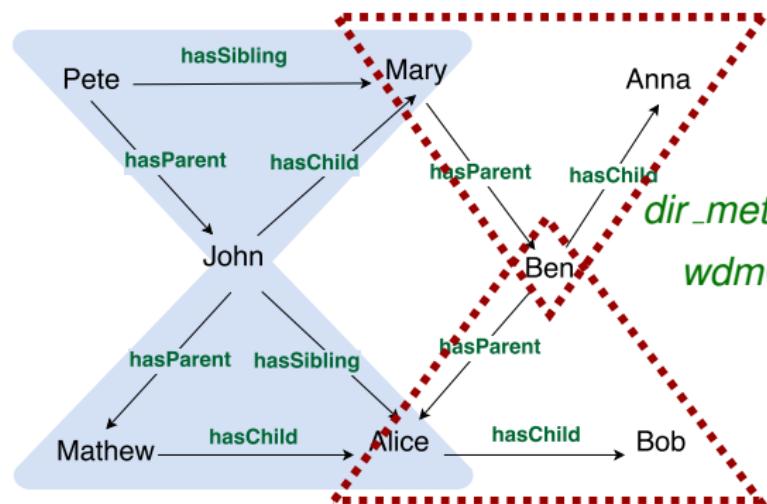
$r : hasSibling(Z, Y) \leftarrow hasChild(X, Y), hasParent(Z, X)$
 $miss(hasSibling, Mary) = 2 \quad \beta = 0.5$

Example with directional metric



$r : hasSibling(Z, Y) \leftarrow hasChild(X, Y), hasParent(Z, X)$
 $miss(hasSibling, Mary) = 2 \quad \beta = 0.5$

Example with directional metric



$$npi(r) = 1 \quad npc(r) = 0$$

$$dir_metric(r) = \frac{1-0}{2 \cdot (1+0)} + 0.5 = 1$$

$$wdm(r) = 0.5 \cdot \frac{2}{4} + 0.5 \cdot 1 = \frac{3}{4}$$

$$\beta = 0.5$$

$r : hasSibling(Z, Y) \leftarrow hasChild(X, Y), hasParent(Z, X)$
 $miss(hasSibling, Mary) = 2 \quad \beta = 0.5$

Motivation

Preliminaries

Rule Learning

Exception-awareness

Incompleteness

Rules from Hybrid Sources

Further Topics

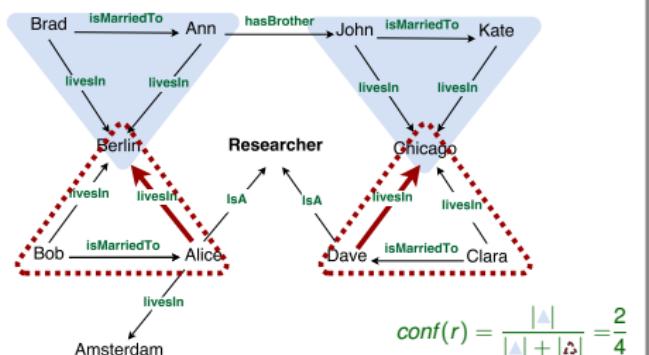
Knowledge Graph Completion

- **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



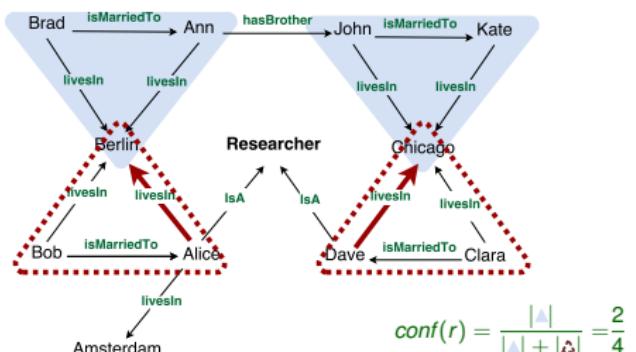
$r : livesIn(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z)$

AMIE [?], [?],
RUMIS [?], CARL [?], 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

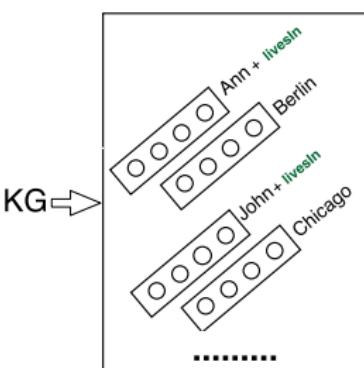
Rule-based approaches



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

AMIE [?], [?],
RUMIS [?], CARL [?], etc.

Statistics-based approaches

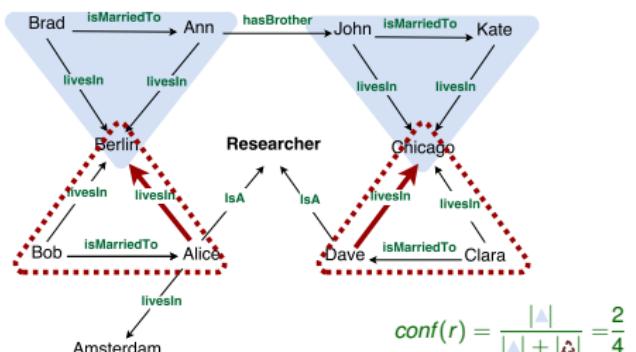


TransE [?], TEKE [?],
RESCAL [?], etc.

Knowledge Graph Completion

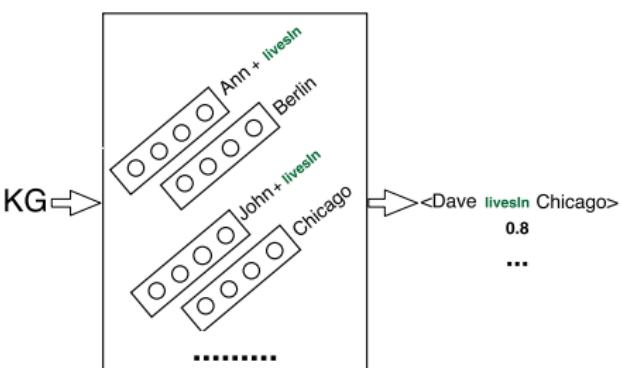
- Given: a KG, i.e., set of $\langle s \ p \ o \rangle$ facts and possibly text
- Find: missing $\langle s \ p \ o \rangle$ facts

Rule-based approaches



AMIE [?], [?],
RUMIS [?], CARL [?], etc.

Statistics-based approaches



TransE [?], TEKE [?],
RESCAL [?], 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.

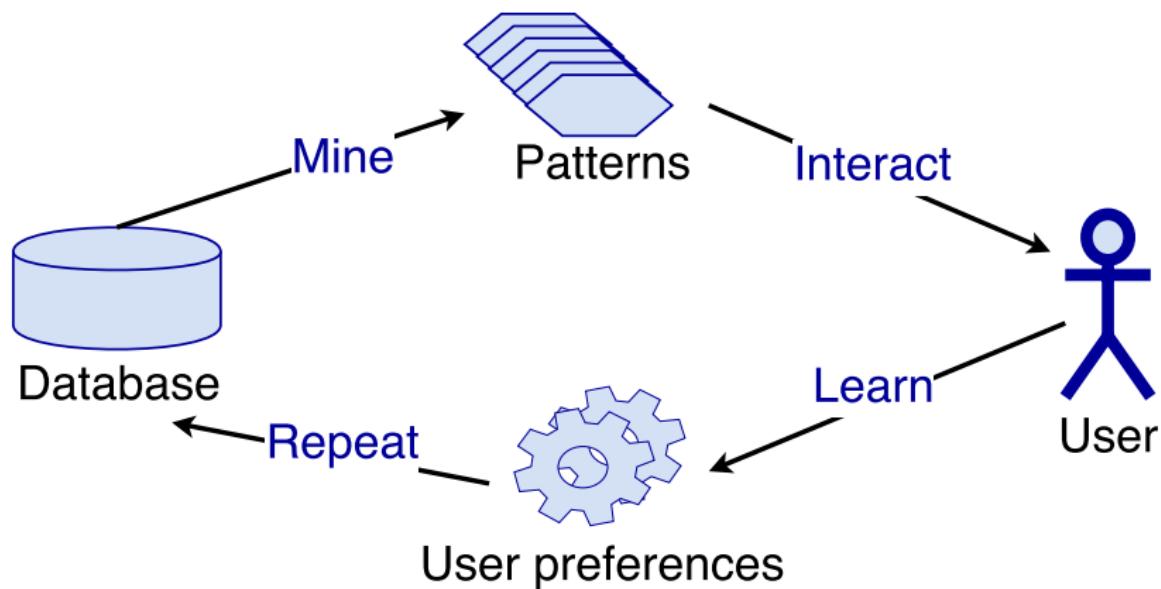
Problem Statement

Feedback-driven rule mining

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

Mine-Interact-Learn-Repeat

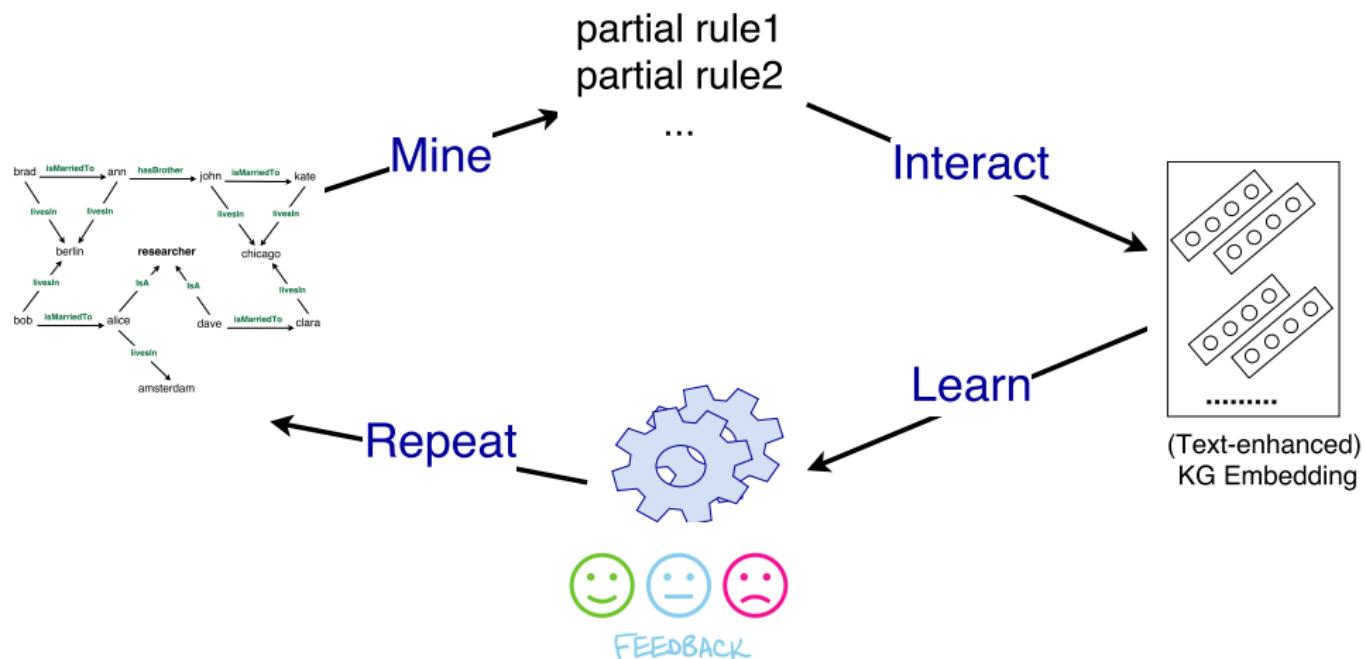
Mimic “mine-interact-learn-repeat” schema [?]



Mine-Interact-Learn-Repeat

Mimic “mine-interact-learn-repeat” schema [?]

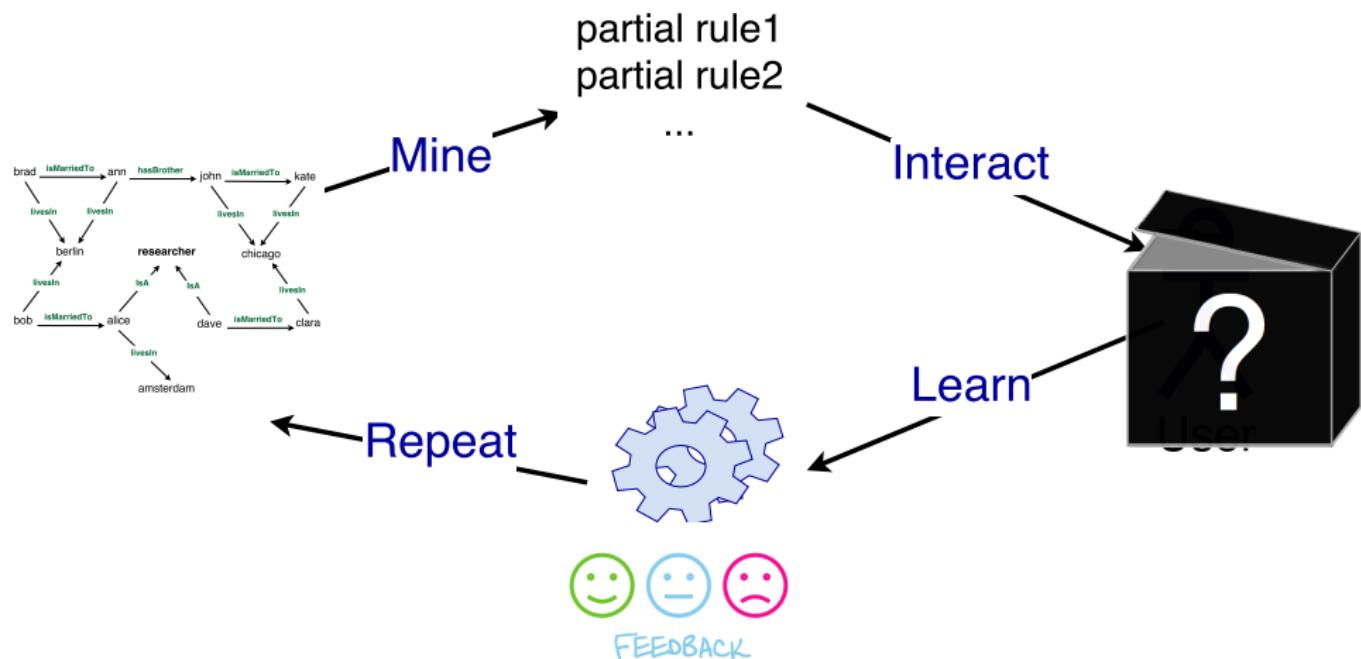
Establish “user-in-the-loop” inspired interaction between the rule mining algorithm and the embedding model



Mine-Interact-Learn-Repeat

Mimic “mine-interact-learn-repeat” schema [?]

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

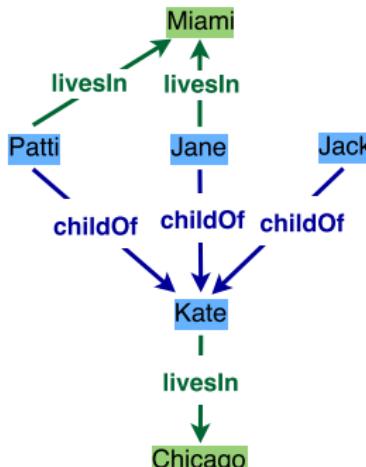
- Q1 (Interact)** What kind of feedback is required/possible to obtain from the “black box” to organize convenient and effective interaction process?
- Q2 (Mine)** How to adapt existing rule mining algorithms to account for feedback?
- Q3 (Learn)** Can anything be learnt from the feedback provided by 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?

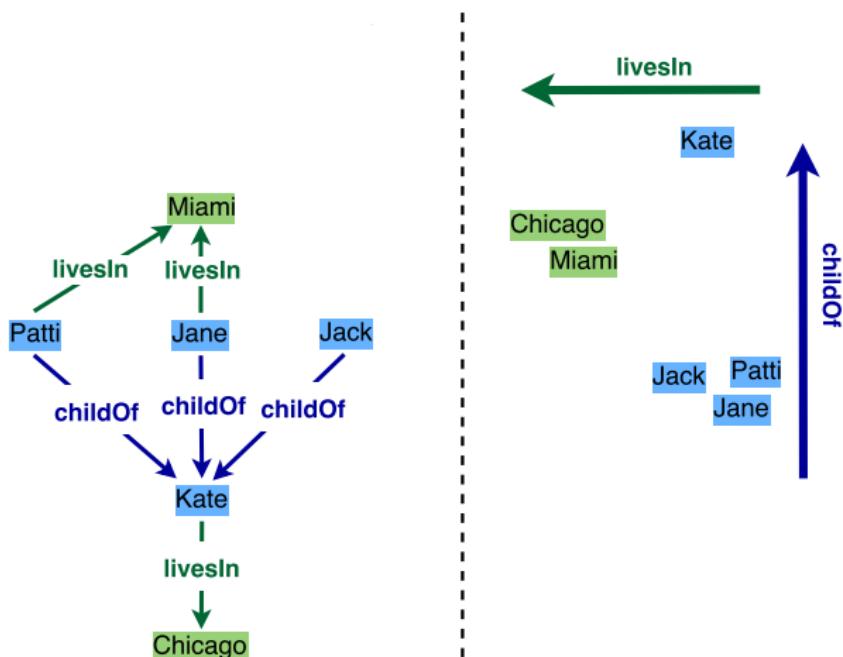
Embedding-based Methods

- **Intuition:** For $\langle s, p, o \rangle$ in KG, find s, p, o such that $s + p \approx o$
- 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



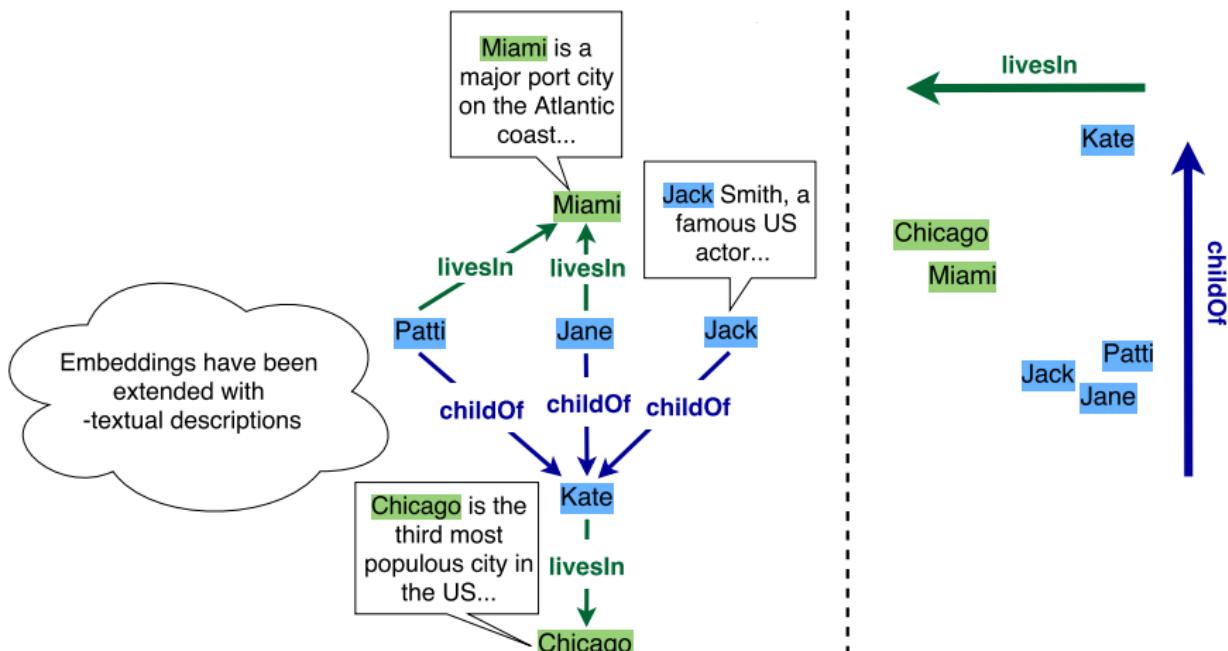
Embedding-based Methods

- **Intuition:** For $\langle s, p, o \rangle$ in KG, find s, p, o such that $s + p \approx o$
- 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



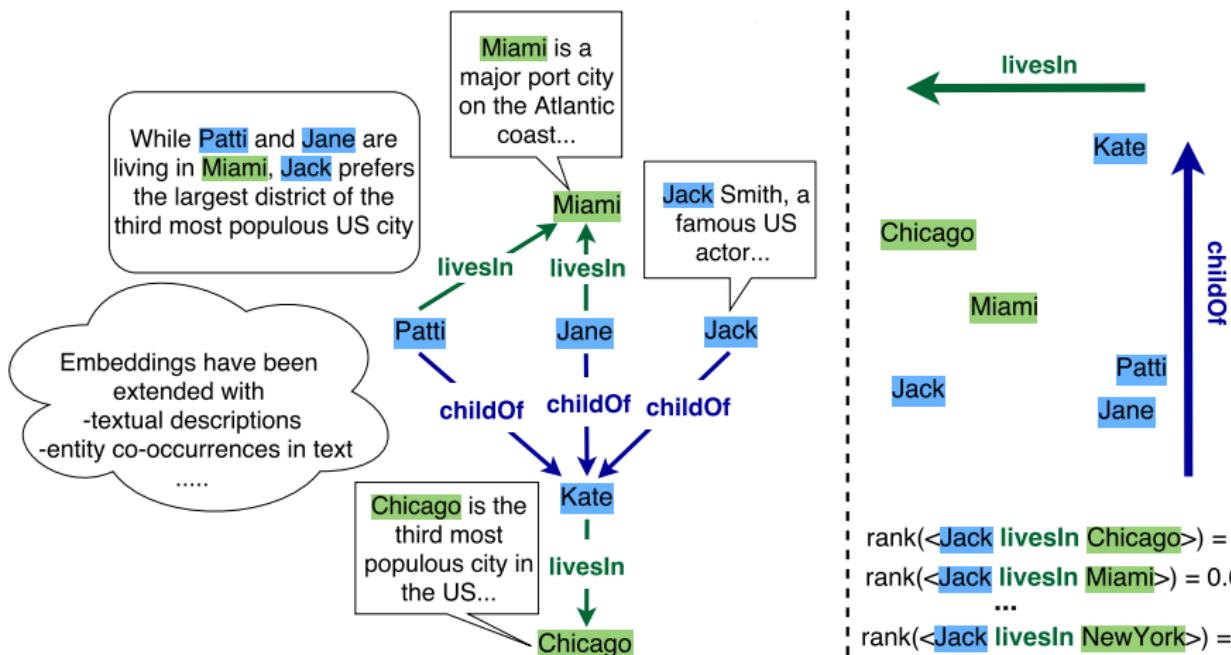
Embedding-based Methods

- **Intuition:** For $\langle s, p, o \rangle$ in KG, find s, p, o such that $s + p \approx o$
- 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 s, p, o such that $s + p \approx o$
- 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

$$\text{rule_mrr}(r) = \frac{1}{|\text{predictions}(r)|} \sum_{\langle s p o \rangle \in \text{predictions}(r)} \text{rank}(\langle s p o \rangle)$$

Q1 (Interact)

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

- rely on average quality of predicted facts

$$\text{rule_mrr}(r) = \frac{1}{|\text{predictions}(r)|} \sum_{\langle s \text{ } p \text{ } o \rangle \in \text{predictions}(r)} \text{rank}(\langle s \text{ } p \text{ } o \rangle)$$

Example

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

- rule predictions: $\langle \text{Jack livesIn NY} \rangle, \langle \text{Mat livesIn Berlin} \rangle$

$$\text{rule_mrr}(r) = \frac{\text{rank}(\langle \text{Jack livesIn NY} \rangle) + \text{rank}(\langle \text{Mat livesIn Berlin} \rangle)}{2}$$

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

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

- combination of mrr with standard rule measures over KG

$$\text{embed_conf}(r) = \lambda * \text{conf}(r) + (1 - \lambda) * \text{rule_mrr}(r),$$

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

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

- combination of mrr with standard rule measures over KG

$$\text{embed_conf}(r) = \lambda * \text{conf}(r) + (1 - \lambda) * \text{rule_mrr}(r),$$

- 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

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)

Tentative 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)

Tentative algorithm steps:

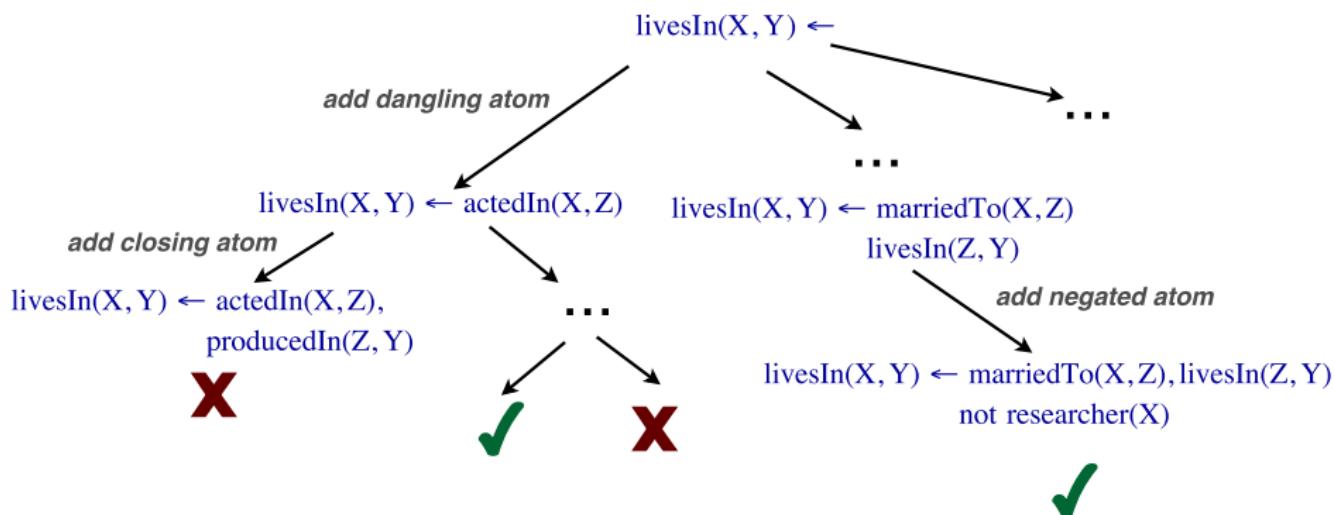
- maintain a rule queue, starting from an empty rule
- for each rule:
 1. process the rule
 - compute statistics: *rule_mrr, embed_conf*
 - filter rules based on statistics and output rule
 2. extend the queue by applying refinement operators

Q2 (Mine)

Tentative 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

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?

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

Evaluation

2 Datasets:

- FB15K: a subset of Freebase with 592K binary facts over 15K entities and 1345 predicates
- Wiki44K: a subset of Wikidata with 250K binary facts over 44K entities and 100 relations

Available KGs: random 80% of the approximated ideal KG

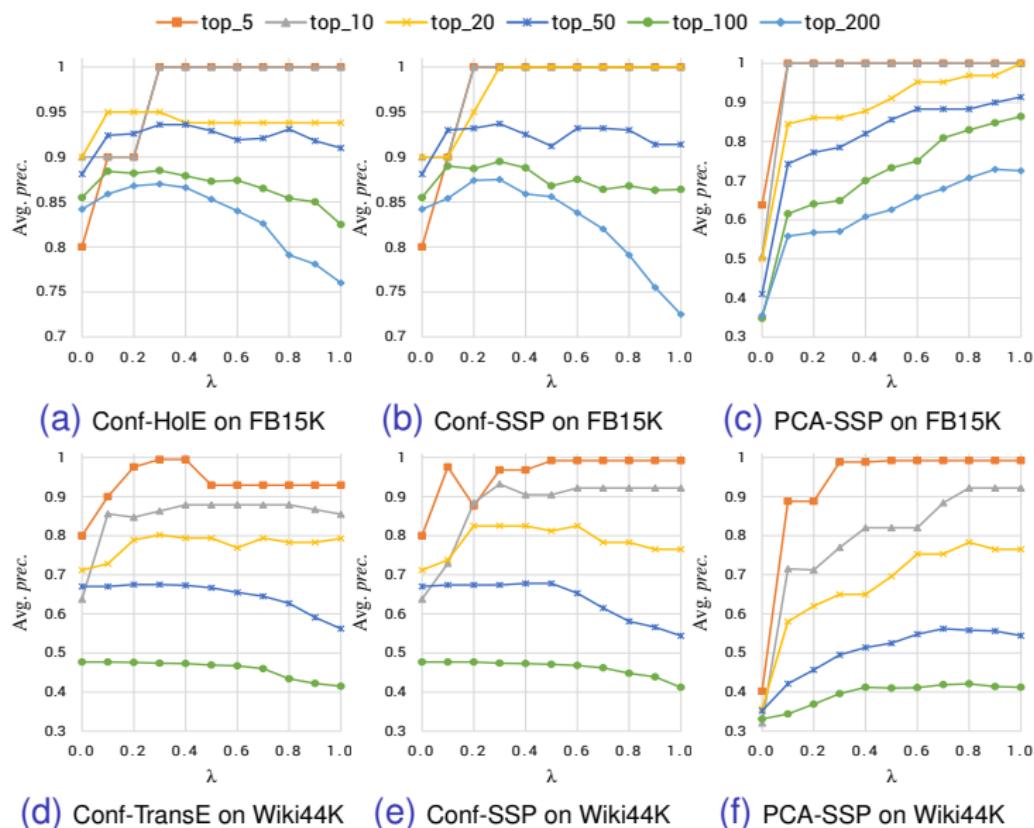
Embedding Models: 2 settings:

- Without additional textual data: TransE model, HolE model
- With entities description text: SSP model

Gold standard: ratio of true facts generated

2 Evaluation settings: *closed world* (CW) and *open world* (OW)

Embedding-Based Hybrid Quality Function



Embedding-Based Hybrid Quality Function

| top- <i>k</i> | FB15K | | | | Wiki44K | | | |
|---------------|-------------------|-------------------|---------------------|---------------------|-------------------|-------------------|---------------------|---------------------|
| | Conf | PCA | Conf-HoIE | Conf-SSP | Conf | PCA | Conf-TransE | Conf-SSP |
| | ($\lambda = 0$) | ($\lambda = 0$) | ($\lambda = 0.3$) | ($\lambda = 0.3$) | ($\lambda = 0$) | ($\lambda = 0$) | ($\lambda = 0.3$) | ($\lambda = 0.3$) |
| 5 | 0.800 | 0.638 | 1.000 | 1.000 | 0.800 | 0.402 | 0.995 | 0.968 |
| 10 | 0.900 | 0.506 | 1.000 | 1.000 | 0.638 | 0.321 | 0.863 | 0.932 |
| 20 | 0.900 | 0.499 | 0.950 | 1.000 | 0.712 | 0.357 | 0.802 | 0.825 |
| 50 | 0.881 | 0.410 | 0.936 | 0.937 | 0.670 | 0.352 | 0.675 | 0.674 |
| 100 | 0.855 | 0.348 | 0.885 | 0.895 | 0.477 | 0.331 | 0.474 | 0.474 |
| 200 | 0.842 | 0.355 | 0.870 | 0.875 | — | — | — | — |

Precision of the *top-k* rules learned using different measures in CW setting.

Horn Rule Learning

| top- <i>k</i> | FB15K | | | | | | Wiki44K | | | | | |
|---------------|----------|-------|-----------|-------|-------|-------|----------|-------|-----------|-------|--------|-------|
| | AMIE-PCA | | AMIE-Conf | | RuLES | | AMIE-PCA | | AMIE-Conf | | RuLES | |
| | Facts | Prec. | Facts | Prec. | Facts | Prec. | Facts | Prec. | Facts | Prec. | Facts | Prec. |
| 20 | 1029 | 0.28 | 82 | 0.63 | 44 | 1.00 | 185 | 0.73 | 91 | 0.95 | 3291 | 0.98 |
| 50 | 1716 | 0.43 | 190 | 0.74 | 186 | 0.92 | 47099 | 0.10 | 3594 | 0.95 | 6154 | 0.88 |
| 100 | 3085 | 0.65 | 255 | 0.78 | 539 | 0.80 | 56831 | 0.20 | 13870 | 0.83 | 13253 | 0.82 |
| 200 | 10586 | 0.62 | 1210 | 0.83 | 1205 | 0.88 | 82288 | 0.39 | 19538 | 0.72 | 20408 | 0.73 |
| 500 | 40050 | 0.51 | 2702 | 0.75 | 7124 | 0.95 | 219264 | 0.35 | 124836 | 0.23 | 128256 | 0.48 |

Precision of the *top-k* rules generated by RuLES and AMIE in OW setting.

Exception-Aware Rule Learning

| <i>top-k</i> | FB15K | | | | Wiki44K | | | |
|--------------|-------|-------|-------|-------|---------|-------|-------|-------|
| | RUMIS | | RuLES | | RUMIS | | RuLES | |
| | Facts | Prec. | Facts | Prec. | Facts | Prec. | Facts | Prec. |
| 20 | 672 | 0.95 | 34 | 0.97 | 5844 | 0.93 | 5640 | 0.93 |
| 50 | 1797 | 0.94 | 158 | 0.99 | 8585 | 0.83 | 13333 | 0.84 |
| 100 | 2672 | 0.94 | 434 | 0.99 | 21081 | 0.76 | 25265 | 0.81 |
| 200 | 4103 | 0.87 | 1155 | 0.96 | 50957 | 0.51 | 43677 | 0.67 |
| 500 | 13439 | 0.76 | 5466 | 0.90 | - | - | - | - |
| <i>top-k</i> | FB15K | | | | Wiki44K | | | |
| | RUMIS | | RuLES | | RUMIS | | RuLES | |
| | Facts | Prec. | Facts | Prec. | Facts | Prec. | Facts | Prec. |
| 20 | 76 | 0.70 | 111 | 0.68 | 63 | 0.47 | 81 | 0.94 |
| 50 | 126 | 0.51 | 435 | 0.74 | 191 | 0.28 | 611 | 0.69 |
| 100 | 183 | 0.43 | 680 | 0.76 | 543 | 0.49 | 1698 | 0.79 |
| 200 | 310 | 0.30 | 1112 | 0.87 | 4861 | 0.40 | 3175 | 0.80 |
| 500 | 1155 | 0.53 | 3760 | 0.59 | - | - | - | - |

Prediction prec. (top) and Revision prec. (bottom) of the *top-k* rules learned by RUMIS and RuLES in OW setting.

Motivation

Preliminaries

Rule Learning

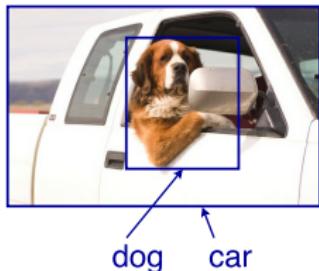
Exception-awareness

Incompleteness

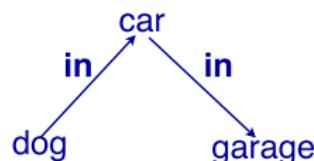
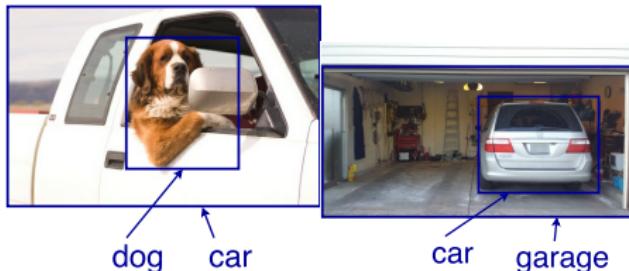
Rules from Hybrid Sources

Further Topics

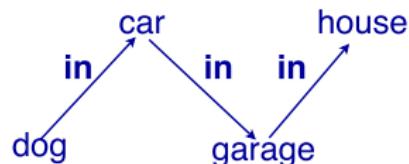
Commonsense Rule Mining from Hybrid Sources



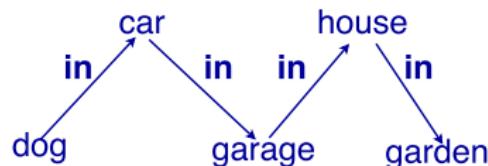
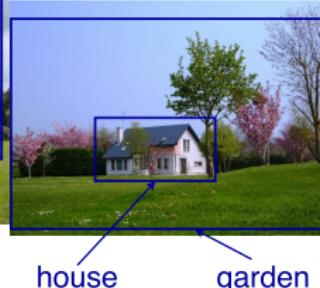
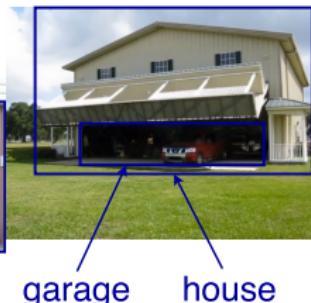
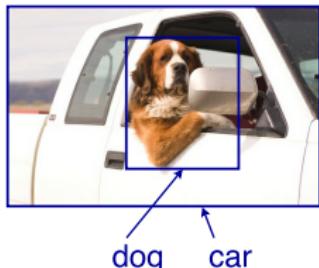
Commonsense Rule Mining from Hybrid Sources



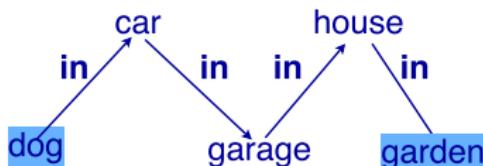
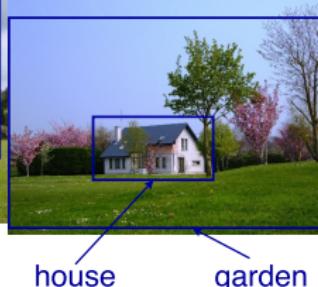
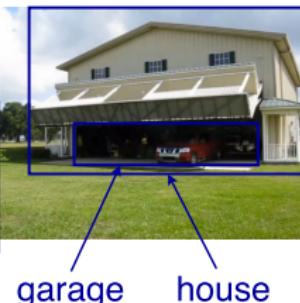
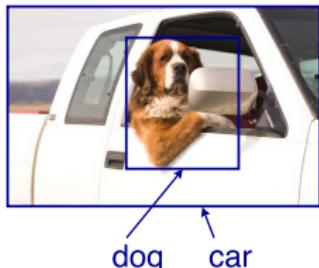
Commonsense Rule Mining from Hybrid Sources



Commonsense Rule Mining from Hybrid Sources

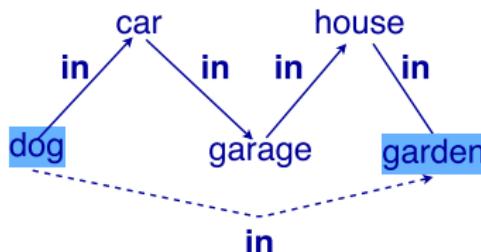
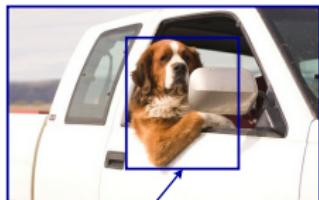


Commonsense Rule Mining from Hybrid Sources

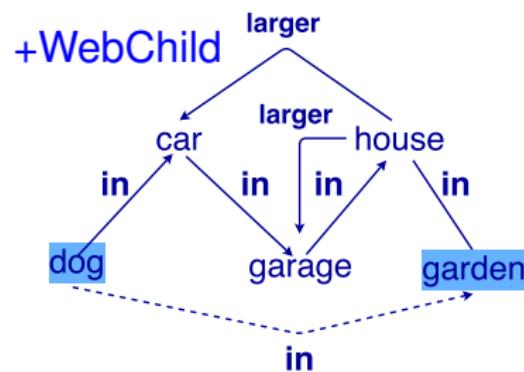
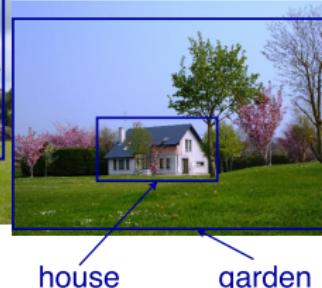
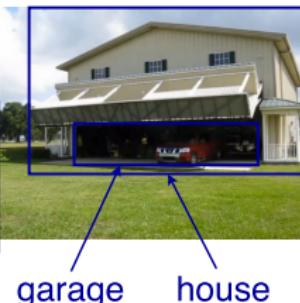
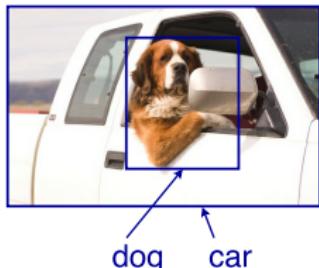


Dog enjoying the sun in the garden

Commonsense Rule Mining from Hybrid Sources

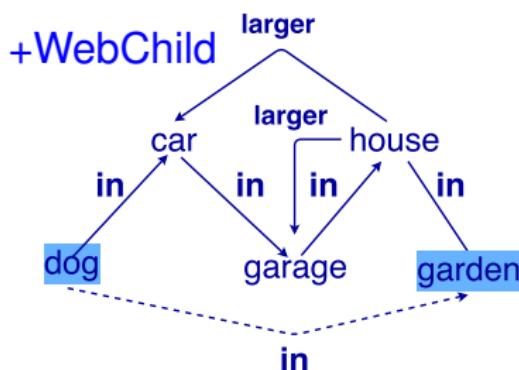
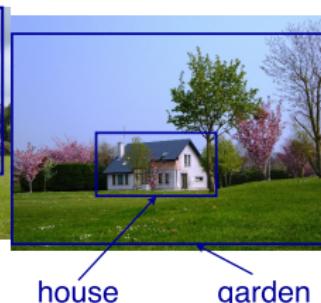
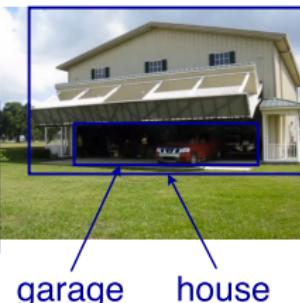
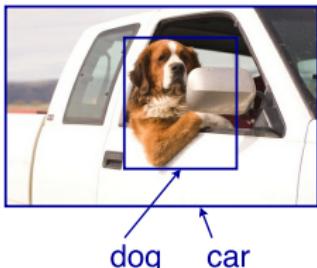


Commonsense Rule Mining from Hybrid Sources



Dog enjoying the sun in the garden

Commonsense Rule Mining from Hybrid Sources



Desired output:

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

...

Commonsense Rule Mining from Hybrid Sources (Text)

- Extracting rules from text challenges
 - Non-canonicalized, ambiguous and infinite set of relations.
 - Noisy facts.
 - Missing negative examples (OWA).

Commonsense Rule Mining from Hybrid Sources (Text)

- **SHERLOCK [?]:** Early attempt to learn rules from open domain text extractions.
- **Golden et. al. [?]:** Utilizes presuppositional discourse patterns (such as statements with but, yet ... etc) to collect conditional knowledge in the form of *if-then*.
- **Dragoni et. al. [?]:** Extracts dependencies between text chunks in legal document such as contract. Then, these dependencies are converted into machine readable rules.
- None of the existing work considered the combination between KGs and Text.

Rule Types

- **Horn:** AMIE [?]

livesIn(Z, Y) ← livesIn(X, Y), marriedTo(X, Z)

Rule Types

- **Horn:** AMIE [?]

livesIn(Z, Y) ← livesIn(X, Y), marriedTo(X, Z)

- **Nonmonotonic:** RUMIS [?], [?]

livesIn(Z, Y) ← livesIn(X, Y), marriedTo(X, Z), not researcher(Z)

Rule Types

- **Horn:** AMIE [?]

livesIn(Z, Y) ← livesIn(X, Y), marriedTo(X, Z)

- **Nonmonotonic:** RUMIS [?], [?]

livesIn(Z, Y) ← livesIn(X, Y), marriedTo(X, Z), not researcher(Z)

- **Numerical:** CARL [?]

hasChild_{≥5}(X) ← hasFather(Y, X), hasSibling_{≥4}(Y)

Rule Types

- **Horn:** AMIE [?]

livesIn(Z, Y) ← livesIn(X, Y), marriedTo(X, Z)

- **Nonmonotonic:** RUMIS [?], [?]

livesIn(Z, Y) ← livesIn(X, Y), marriedTo(X, Z), not researcher(Z)

- **Numerical:** CARL [?]

hasChild_{≥5}(X) ← hasFather(Y, X), hasSibling_{≥4}(Y)

- **Disjunctive:**

male(Y) ∨ female(Y) ← hasParent(X, Y)

Rule Types

- **Horn:** AMIE [?]

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

- **Nonmonotonic:** RUMIS [?], [?]

$livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z), \text{not researcher}(Z)$

- **Numerical:** CARL [?]

$hasChild_{\geq 5}(X) \leftarrow hasFather(Y, X), hasSibling_{\geq 4}(Y)$

- **Disjunctive:**

$male(Y) \vee female(Y) \leftarrow hasParent(X, Y)$

- **Existential:**

$\exists Y hasParent(X, Y) \leftarrow person(X)$

Rule Types

- **Horn:** AMIE [?]

livesIn(Z, Y) ← livesIn(X, Y), marriedTo(X, Z)

- **Nonmonotonic:** RUMIS [?], [?]

livesIn(Z, Y) ← livesIn(X, Y), marriedTo(X, Z), not researcher(Z)

- **Numerical:** CARL [?]

hasChild_{≥5}(X) ← hasFather(Y, X), hasSibling_{≥4}(Y)

- **Disjunctive:**

male(Y) ∨ female(Y) ← hasParent(X, Y)

- **Existential:**

∃Y hasParent(X, Y) ← person(X)

- **Temporal constraints:**

⊥ ← bornIn(X, Y), after(Y, Z), studied(X, Z)

References I

-  Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek.
Fast rule mining in ontological knowledge bases with AMIE+.
VLDB J., 24(6):707–730, 2015.
-  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.