Rule Induction and Reasoning over Knowledge Graphs

26.09.2018

max planck institut informatik

Outline

Knowledge Graphs

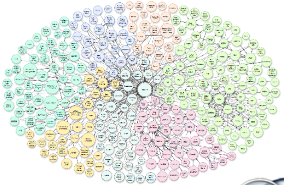
“Semantically enriched machine processable data”

2018

Personal

External

yago
select knowledge



WIKIDATA

NELL

Knowledge Graph

Semantic Web Search

Google winner of Australian Open 2018

Roger Federer

Tennis player

rogerfederer.com

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

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

Height: 1.85 m

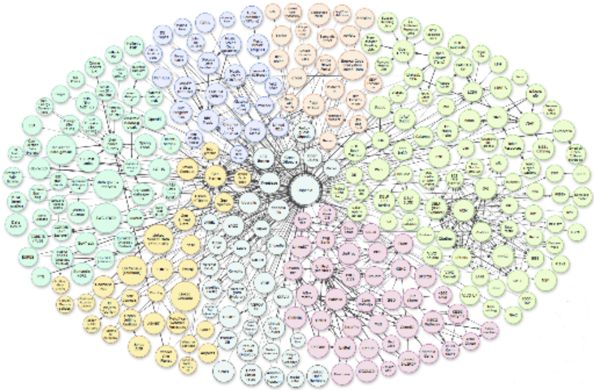
Weight: 85 kg

Spouse: Mirka Federer (m. 2009)

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

Semantic Web Search

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



Roger Federer

Tennis player

rogerfederer.com

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

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

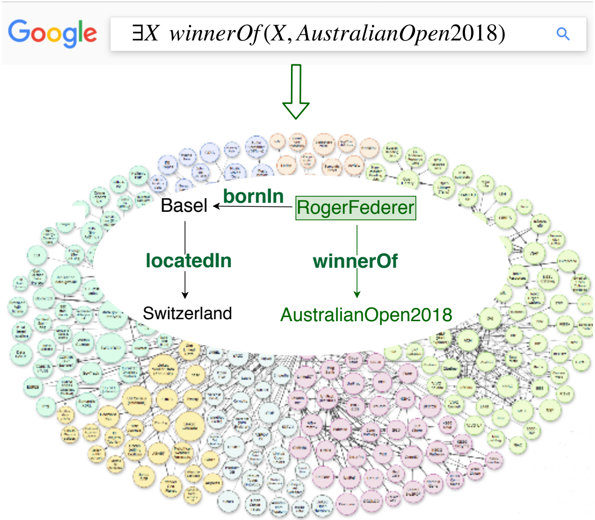
Height: 1.85 m

Weight: 85 kg

Spouse: Mirka Federer (m. 2009)

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

Semantic Web Search



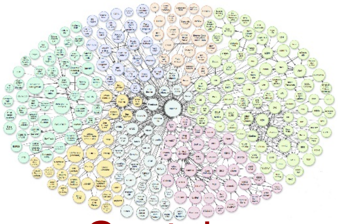
Problem: Inconsistency

Knowledge Graph

Information Extraction

Web pages

Difficult

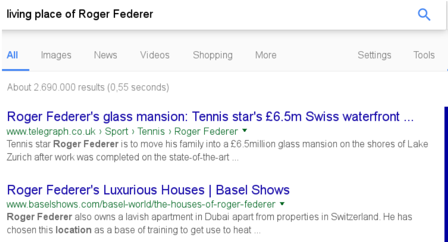


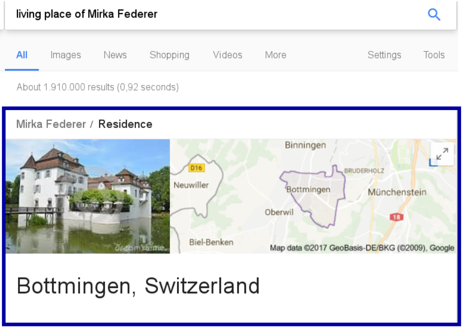
Noisy

Contains mistakes

Problem: Incompleteness

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





What if we had rules?

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

Married people live together

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

What if we had rules?

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

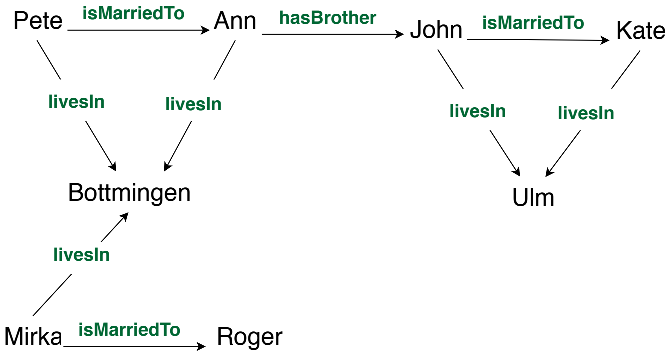
$marriedTo(mirka, roger)$ — *Mirka is married to Roger*

$livesIn(mirka, bottmingen)$ — *Mirka lives in Bottmingen*

$livesIn(roger, bottmingen)$ — *Roger lives in Bottmingen*

But where can one get such rules from?

Extracting Rules from Knowledge Graphs



$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

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

In this Tutorial..

Important problems of KGs:

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

In this Tutorial..

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

Horn Rules

Def.: Horn rule

A Horn rules r is an expression of the form

$$a \leftarrow b_1, \ldots, b_m, \quad (1)$$

where a, $b_1, \ldots, b_m$ are atoms.

- a is the head of the rule
- $b_1, \ldots, b_m$ is the body of the rule.
- If $m = 0$, the rule is a fact (written shortly a)

Intuitively, (1) can be seen as material implication

$$\forall \mathbf{x}\ b_1 \wedge \cdots \wedge b_m \rightarrow a, \text{ where } \mathbf{x}$$

is the list of all variables occurring in (1).

Example

Herbrand Semantics

Def.: **Herbrand universe, base, interpretation**

- Given a logic program P, the Herbrand universe of P, $HU(P)$, is the set of all terms which can be formed from constants and functions symbols in P (resp., the vocabulary Φ of P, if explicitly known).
- The Herbrand base of P, $HB(P)$, is the set of all ground atoms which can be formed from predicates and terms $t \in HU(P)$.

Herbrand Semantics

Def.: Herbrand universe, base, interpretation

- Given a logic program P, the Herbrand universe of P, $HU(P)$, is the set of all terms which can be formed from constants and functions symbols in P (resp., the vocabulary Φ of P, if explicitly known).
- The Herbrand base of P, $HB(P)$, is the set of all ground atoms which can be formed from predicates and terms $t \in HU(P)$.
- A (Herbrand) interpretation is a first-order interpretation $I = (D, \cdot^I)$ of the vocabulary with domain $D = HU(P)$ where each term $t \in HU(P)$ is interpreted by itself, i.e., $t^I = t$.

Herbrand Semantics

Def.: Herbrand universe, base, interpretation

- Given a logic program P, the Herbrand universe of P, $HU(P)$, is the set of all terms which can be formed from constants and functions symbols in P (resp., the vocabulary Φ of P, if explicitly known).
- The Herbrand base of P, $HB(P)$, is the set of all ground atoms which can be formed from predicates and terms $t \in HU(P)$.
- A (Herbrand) interpretation is a first-order interpretation $I = (D, \cdot^I)$ of the vocabulary with domain $D = HU(P)$ where each term $t \in HU(P)$ is interpreted by itself, i.e., $t^I = t$.
- I is identified with the set $\{ \, p(t_1, \ldots, t_n) \in HB(P) \mid \langle t_1^I, \ldots, t_n^I \rangle \in p^I \, \}$.

Herbrand Semantics

Def.: Herbrand universe, base, interpretation

- Given a logic program P, the Herbrand universe of P, $HU(P)$, is the set of all terms which can be formed from constants and functions symbols in P (resp., the vocabulary Φ of P, if explicitly known).
- The Herbrand base of P, $HB(P)$, is the set of all ground atoms which can be formed from predicates and terms $t \in HU(P)$.
- A (Herbrand) interpretation is a first-order interpretation $I = (D, \cdot^I)$ of the vocabulary with domain $D = HU(P)$ where each term $t \in HU(P)$ is interpreted by itself, i.e., $t^I = t$.
- I is identified with the set $\{\ p(t_1, \ldots, t_n) \in HB(P) \mid \langle t_1^I, \ldots, t_n^I \rangle \in p^I\ \}$.

Informally, a (Herbrand) interpretation can be seen as a set denoting which ground atoms are true in a given scenario.

Herbrand Semantics

Def.: Herbrand universe, base, interpretation

- Given a logic program P, the Herbrand universe of P, $HU(P)$, is the set of all terms which can be formed from constants and functions symbols in P (resp., the vocabulary Φ of P, if explicitly known).
- The Herbrand base of P, $HB(P)$, is the set of all ground atoms which can be formed from predicates and terms $t \in HU(P)$.
- A (Herbrand) interpretation is a first-order interpretation $I = (D, \cdot^I)$ of the vocabulary with domain $D = HU(P)$ where each term $t \in HU(P)$ is interpreted by itself, i.e., $t^I = t$.
- I is identified with the set $\{ p(t_1, \ldots, t_n) \in HB(P) \mid \langle t_1^I, \ldots, t_n^I \rangle \in p^I \}$.

Informally, a (Herbrand) interpretation can be seen as a set denoting which ground atoms are true in a given scenario.

Example

Grounding Example

Herbrand Models

Def.: **Herbrand models**

An interpretation I is a (Herbrand) model of

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

Proposition

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

Example

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.

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 $J \subset I$.

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 $J \subset I$.

Theorem

Every positive logic program P has a single minimal model (called the least model), denoted $LM(P)$.

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 $J \subset I$.

Theorem

Every positive logic program P has a single minimal model (called the least model), denoted $LM(P)$.

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

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.

Example

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

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

Nonmonotonic Rules

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

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

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

not **is different from $\neg$!**

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

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

Answer Set Semantics

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

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

Answer Set Semantics

Evaluation of ASP programs is model-based[1], it consists of 2 steps:

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

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

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

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:

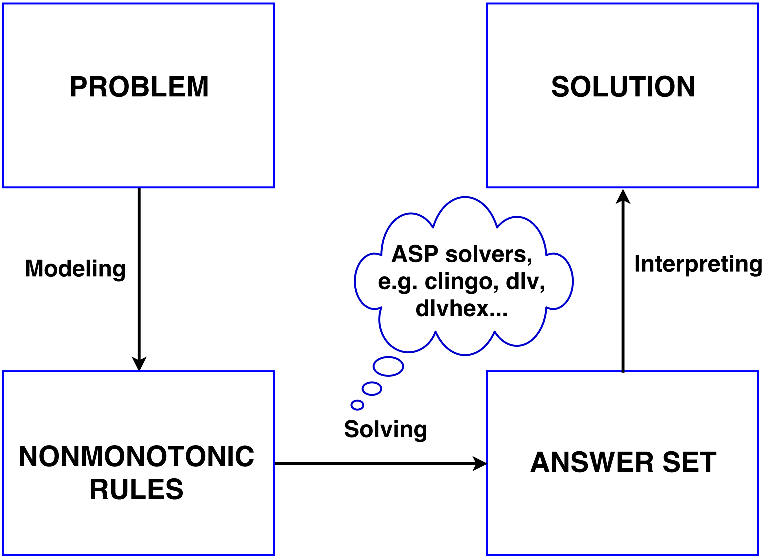
1. Grounding: substitute all variables with constants in all possible ways
2. Solving: compute a minimal model (answer set) *I* satisfying all rules

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

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

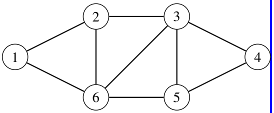
Nonmonotonicity: adding facts might lead to loss of consequences!

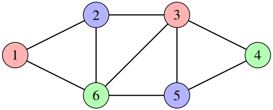
ASP: Declarative Programming Paradigm



Declarative Programming Example

Graph 3-colorability





Modeling

$node(1 \ldots 6)$; $edge(1, 2)$; ...

$col(V, red) \leftarrow not\ col(V, blue), not\ col(V, green), node(V)$;

$col(V, green) \leftarrow not\ col(V, blue), not\ col(V, red), node(V)$;

$col(V, blue) \leftarrow not\ col(V, green), not\ col(V, red), node(V)$;

$\perp \leftarrow col(V, C), col(V, C'), C \neq C'$;

$\perp \leftarrow col(V, C), col(V', C), edge(V, V')$

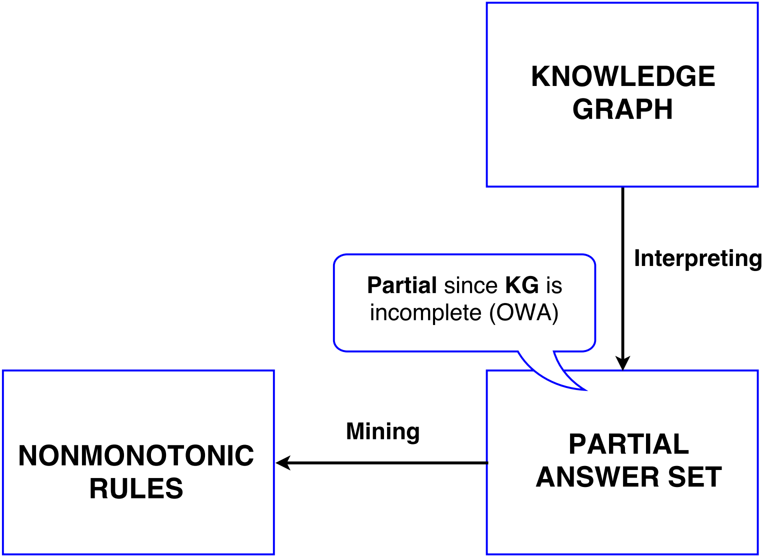
Interpreting

NONMONOTONIC RULES

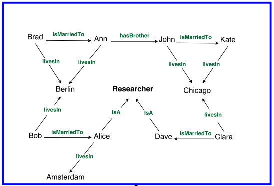
Solving

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

Rule Mining


KNOWLEDGE GRAPH
Interpreting
Partial since **KG** is incomplete (OWA)
PARTIAL ANSWER SET
Mining
NONMONOTONIC RULES

Rule Mining



Interpreting

isMarriedTo(*brad*, *ann*);
isMarriedTo(*john*, *kate*);
isMarriedTo(*bob*, *alice*);
isMarriedTo(*clara*, *dave*);
livesIn(*brad*, *berlin*;
...
researcher(*alice*);
researcher(*dave*)

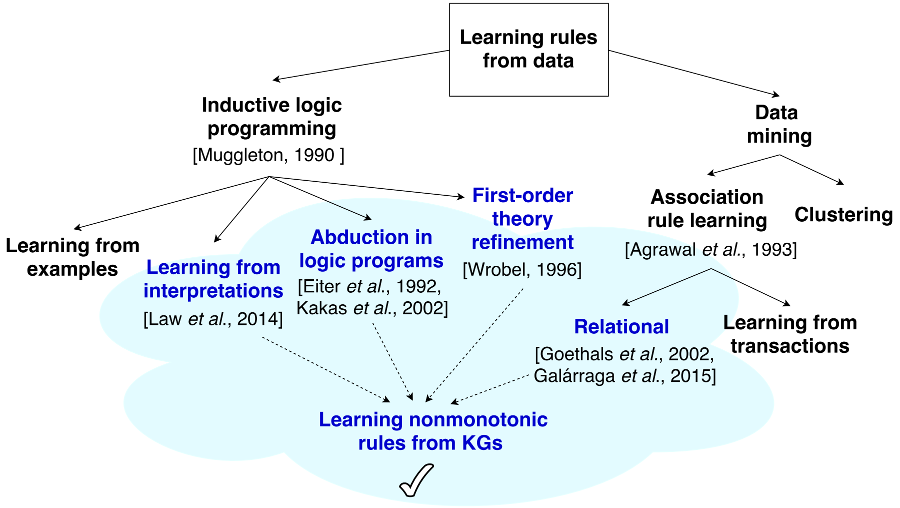
Mining

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

Rule Mining from KGs

Goal: learn nonmonotonic rules from KG

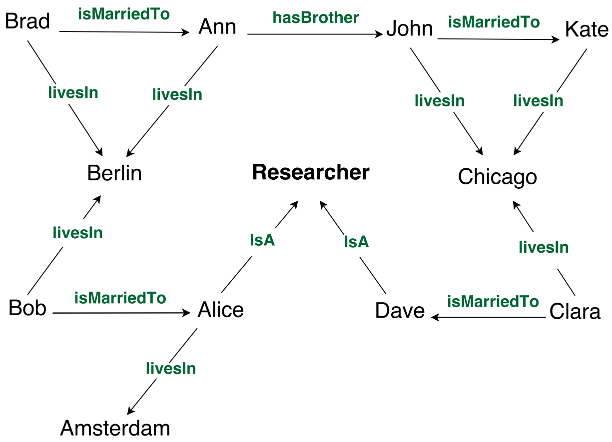
Approach: revise association rules learned using data mining methods



Itemset Mining

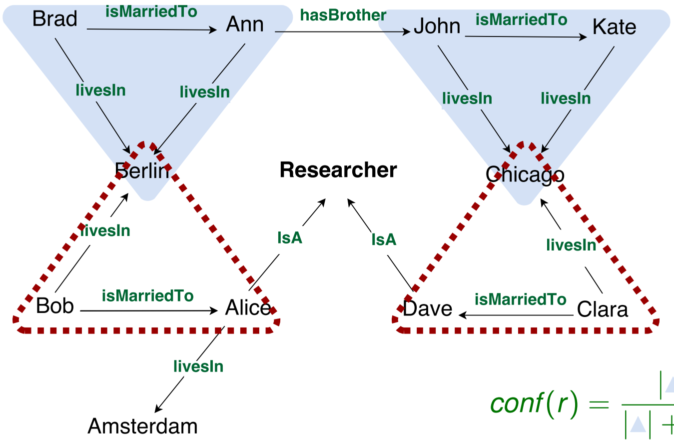
Relational Pattern Mining

Horn Rule Mining



Horn Rule Mining

Horn rule mining for KG completion [?]

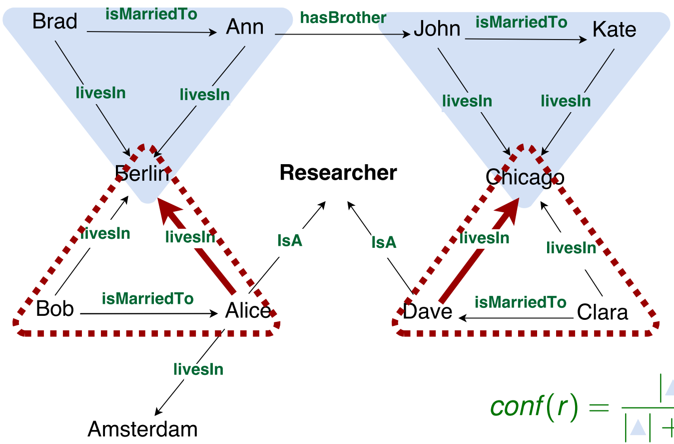


$$conf(r) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

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

Horn Rule Mining

Horn rule mining for KG completion [?]

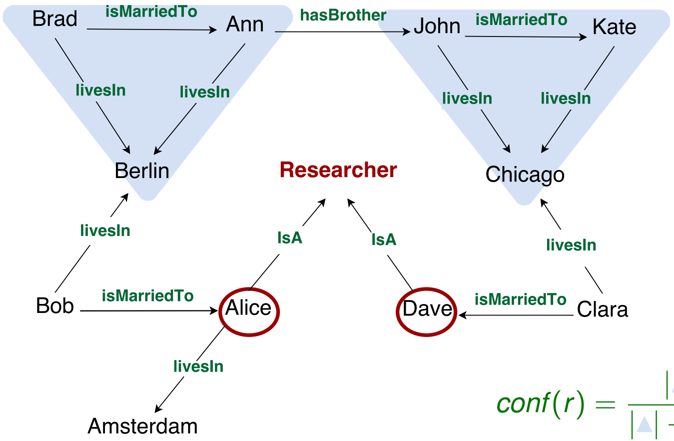


$$conf(r) = \frac{|\triangle|}{|\triangle| + |\triangle|} = \frac{2}{4}$$

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

Nonmonotonic Rule Mining

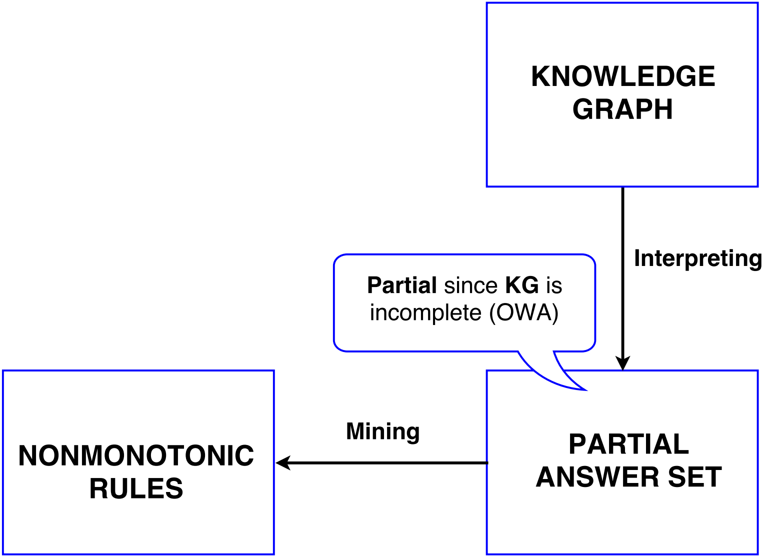
Nonmonotonic rule mining from KGs: OWA is a challenge!



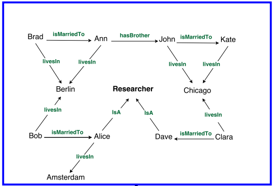
$$conf(r) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = 1$$

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

Nonmonotonic Rule Mining



Nonmonotonic Rule Mining



Interpreting

$isMarriedTo(brad, ann);$
$isMarriedTo(john, kate);$
$isMarriedTo(bob, alice);$
$isMarriedTo(clara, dave);$
$livesIn(brad, berlin;$
$\ldots$
$researcher(alice);$
$researcher(dave)$

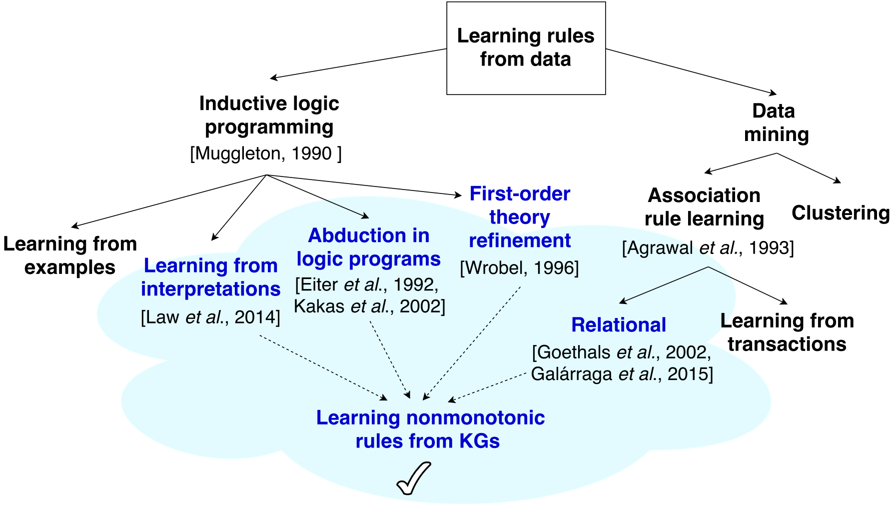
Mining

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

Nonmonotonic Rule Mining from KGs

Goal: learn nonmonotonic rules from KG

Approach: revise association rules learned using data mining methods

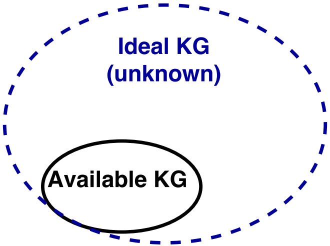


Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG

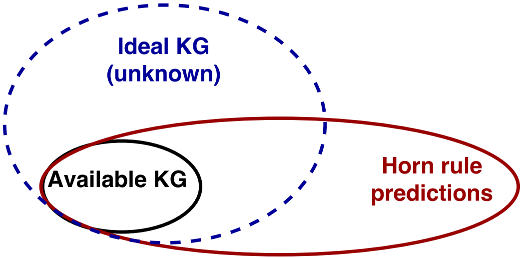


Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set

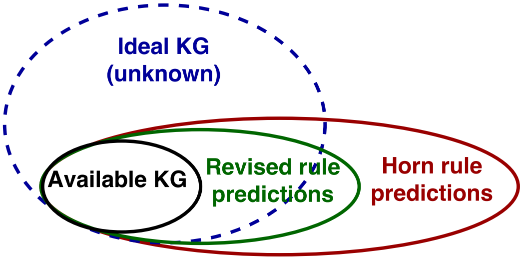


Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set



Find:

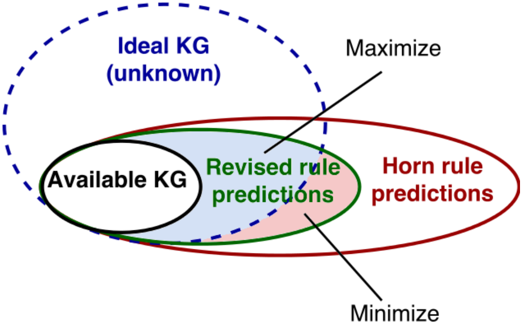
- Nonmonotonic revision of Horn rule set

Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set



Find:

- Nonmonotonic revision of Horn rule set
  with better predictive quality

Avoid Data Overfitting

How to distinguish exceptions from noise?

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

Avoid Data Overfitting

How to distinguish exceptions from noise?

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

Avoid Data Overfitting

How to distinguish exceptions from noise?

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

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

Avoid Data Overfitting

How to distinguish exceptions from noise?

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

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

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

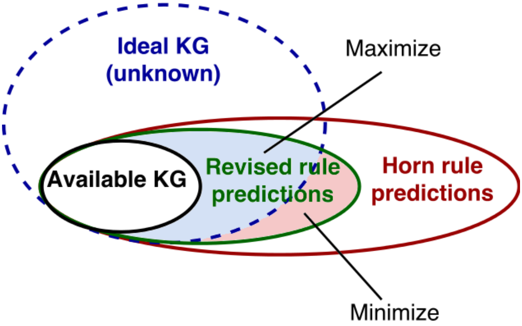
Intuition: Rules with good exceptions should make few conflicting predictions

Horn Theory Revision

Quality-based Horn Theory Revision

Given:

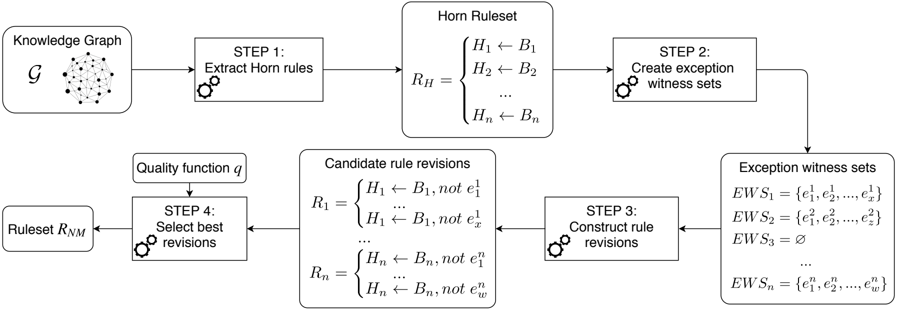
- Available KG
- Horn rule set



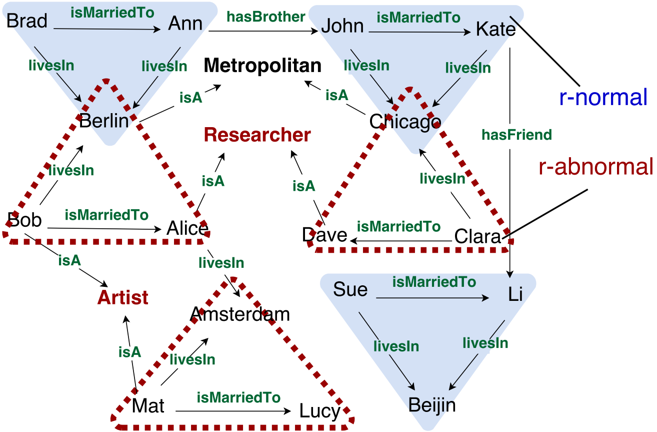
Find:

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

Approach Description



Exception Candidates



$$r\colon livesIn(X,Z) \leftarrow isMarriedTo(Y,X)\,,\, livesIn(Y,Z) \quad \begin{Bmatrix} not\ \ researcher(X) \\ not\ \ artist(Y) \end{Bmatrix}$$

Exception Ranking

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

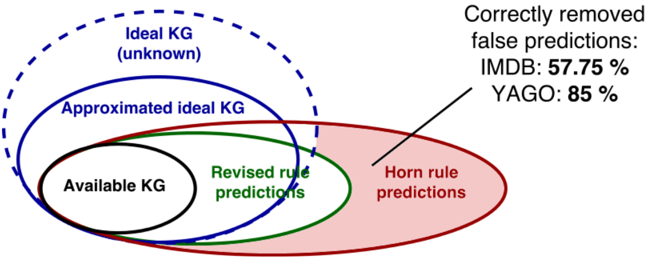
Finding globally best revision is expensive, exponentially many candidates!

- 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

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*

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

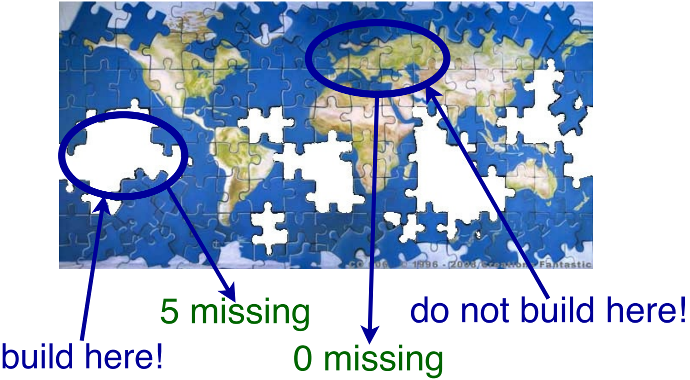
$r_1 : writtenBy(X, Z) \leftarrow hasPredecessor(X, Y), writtenBy(Y, Z), not\ american\_film(X)$

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

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

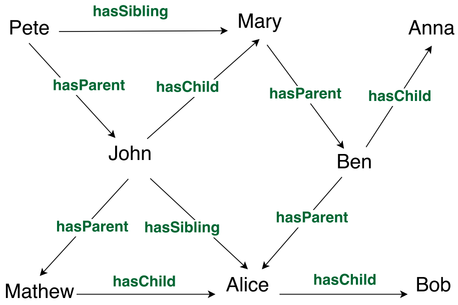
Completeness-aware Rule Mining

- Exploit cardinality meta-data [?] in rule mining
  *John has **5** children, Mary is a citizen of **2** countries*



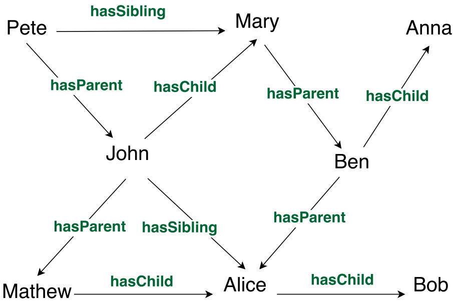
In Proc. ISWC 2017, Completeness-aware Rule Learning from Knolwegde Graphs. T. Pellissier-Tanon, D. Stepanova, S. Razniewski, P. Mirza, G. Weikum, 2018.

Reasonable Rules



Reasonable Rules

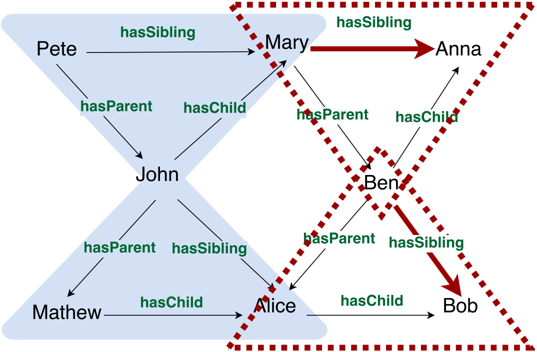
People with the same parents are likely siblings



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

Reasonable Rules

✓ *People with the same parents are likely siblings*

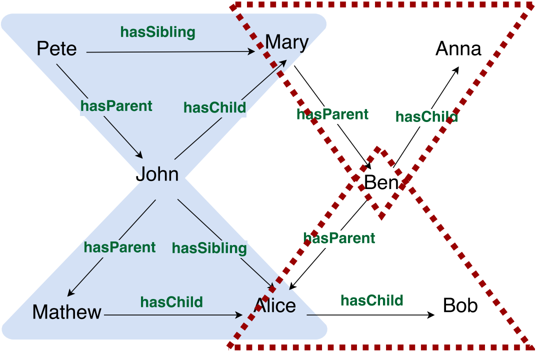


$r_1: \; hasSibling(Z, Y) \leftarrow hasChild(X, Y), hasParent(Z, X)$

Reasonable Rules

✓ *People with the same parents are likely siblings*

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



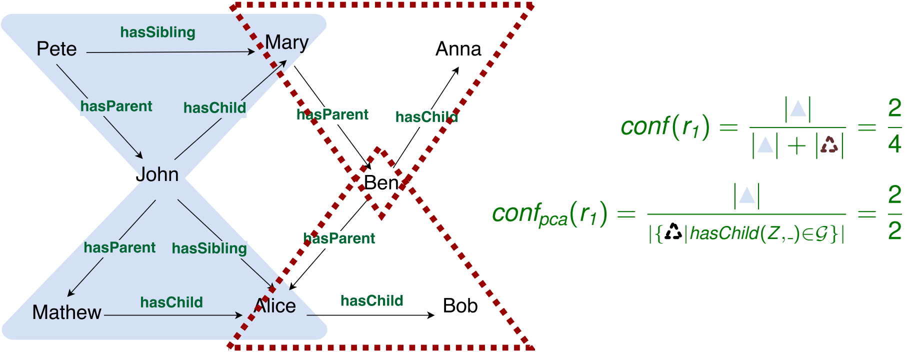
$$conf(r_1) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

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

Reasonable Rules

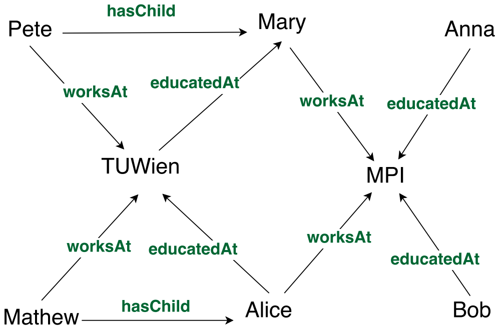
✓ *People with the same parents are likely siblings*

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



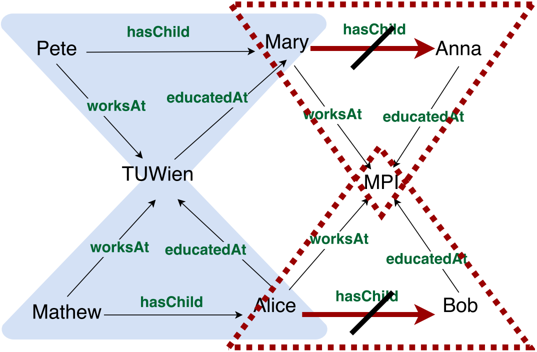
r_1 : *hasSibling*$(Z, Y) \leftarrow$ *hasChild*(X, Y), *hasParent*(Z, X)

Erroneous Rules due to Data Bias



Erroneous Rules due to Data Bias

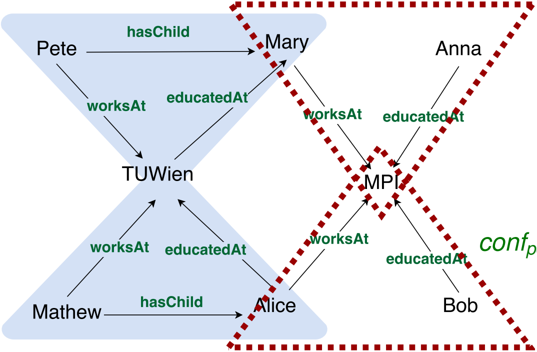
People working and studying at the same institute are likely relatives



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

Erroneous Rules due to Data Bias

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



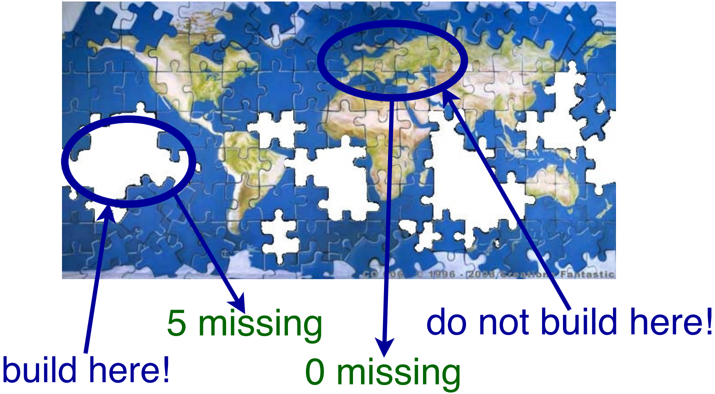
$$conf(r_2) = \frac{|\blacktriangle|}{|\blacktriangle| + |\text{♻}|} = \frac{2}{4}$$

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

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

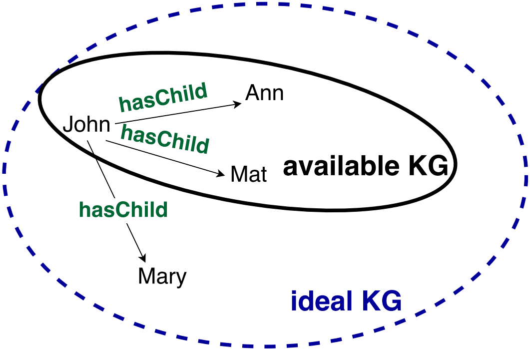
Exploiting Meta-data in Rule Learning

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



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 $incomplete(p, s)$

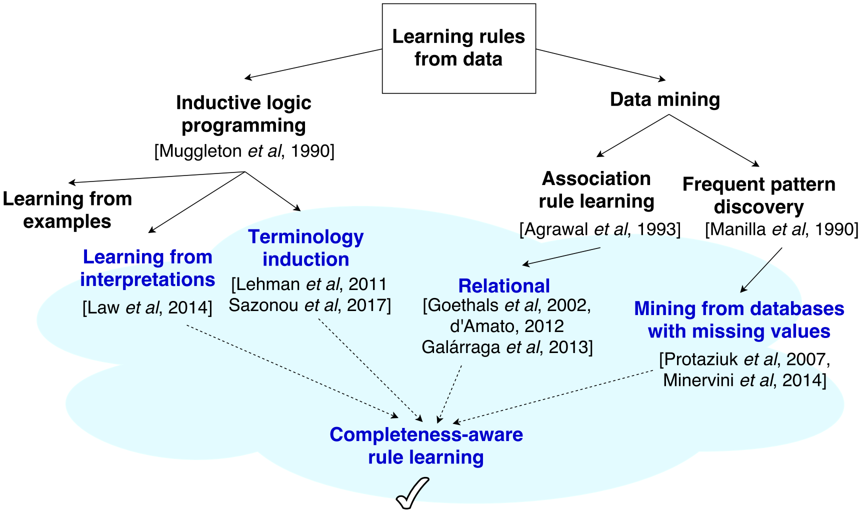


$num(hasChild, john) = 3$
$miss(hasChild, john) = 1$
$incomplete(hasChild, john)$

Cardinality Constraints on Edges

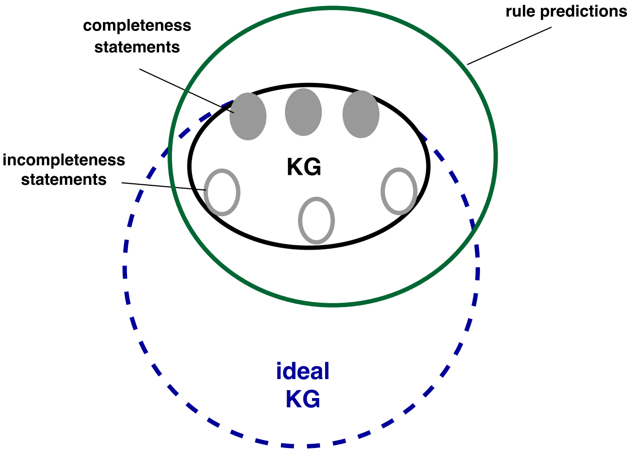
- Mining cardinality assertions from the Web **[?]**
  - *"... John has **2** children ..."*
- Estimating recall of KGs by crowd sourcing **[?]**
  - *20 % of Nobel laureates in physics are missing*
- Predicting completeness in KGs **[?]**
  - Add $\mathit{complete}(\mathit{john}, \mathit{hasChild})$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

Related Work



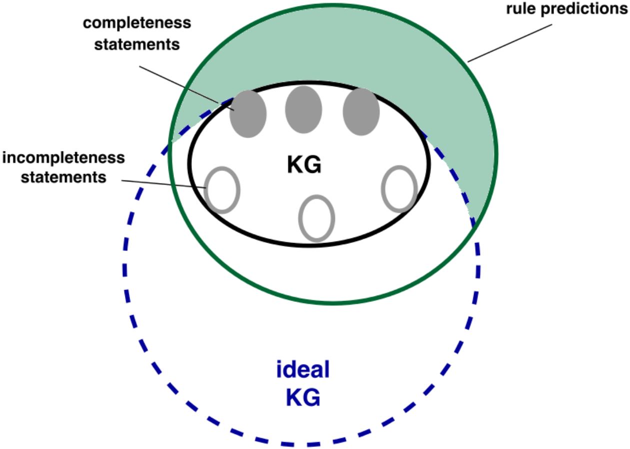
Prediction Post-processing

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



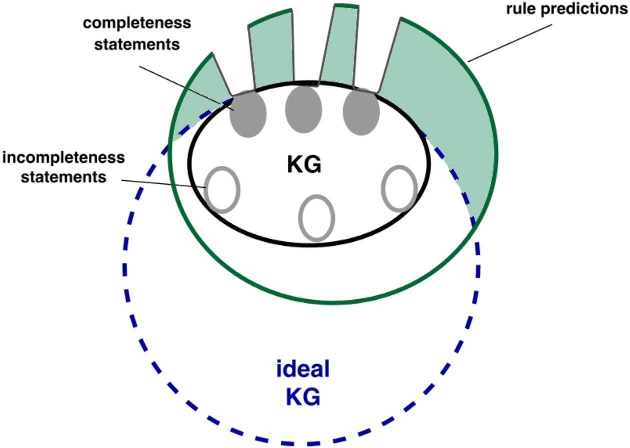
Prediction Post-processing

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



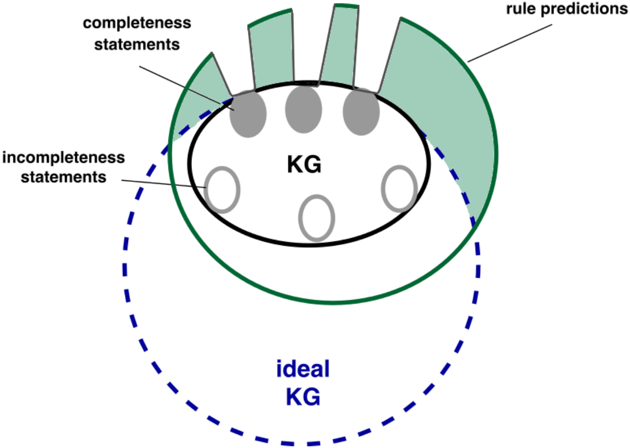
Prediction Post-processing

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

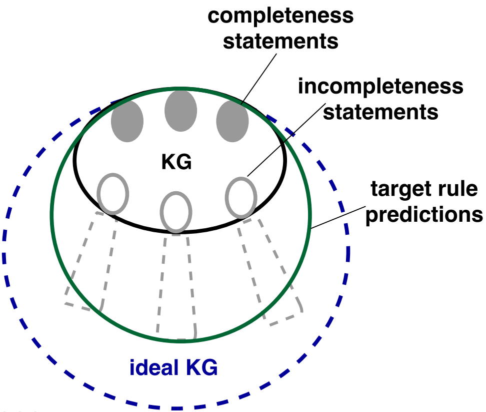


Rules might be still **erroneous**.. What about other incorrect predictions?

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

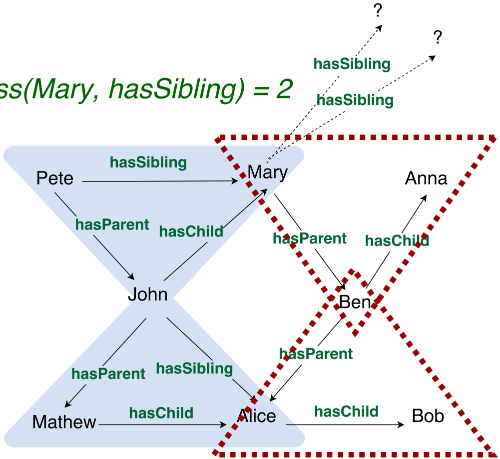
$npi(r)$: number of facts added to incomplete areas by r

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

Rule Predictions

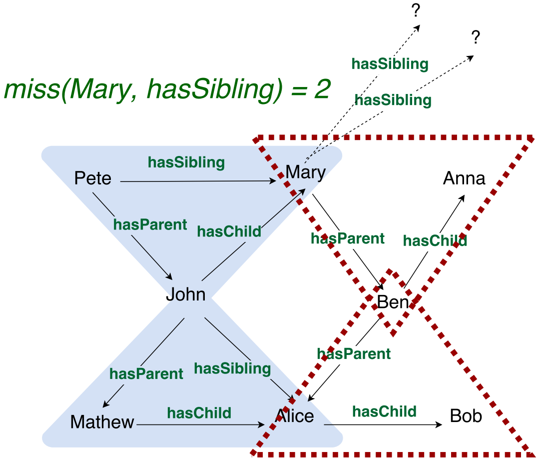
npi(*r*): number of facts added to incomplete areas by *r*
npc(*r*): number of facts added to complete areas by *r*

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

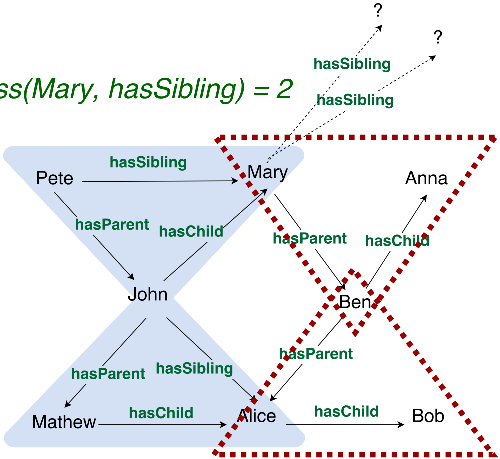
$conf_{comp}$: do not penalize rules that predict new facts in incomplete areas

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

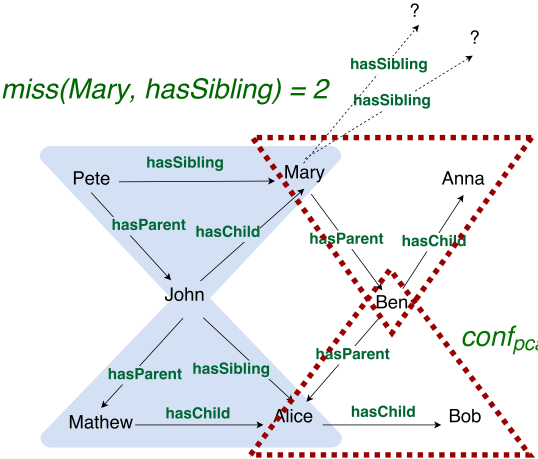
- Generalizes standard confidence ($miss(r) = 0$)
- Generalizes PCA confidence ($miss(r) \in \{0, +\infty\}$)

Completeness Confidence Example 1

miss(Mary, hasSibling) = 2



Completeness Confidence Example 1



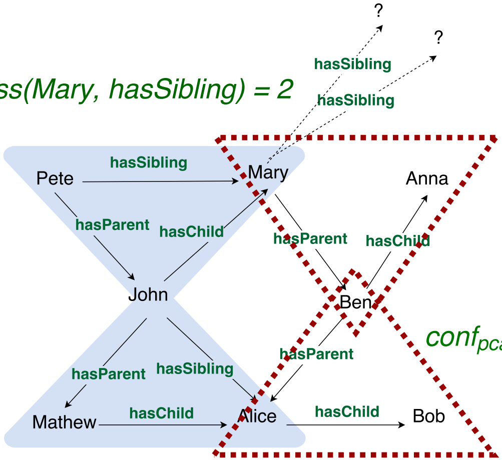
$$conf(r_1) = \frac{|\blacktriangle|}{|\blacktriangle| + |\vartriangle|} = \frac{2}{4}$$

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

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

Completeness Confidence Example 1

miss(Mary, hasSibling) = 2



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

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

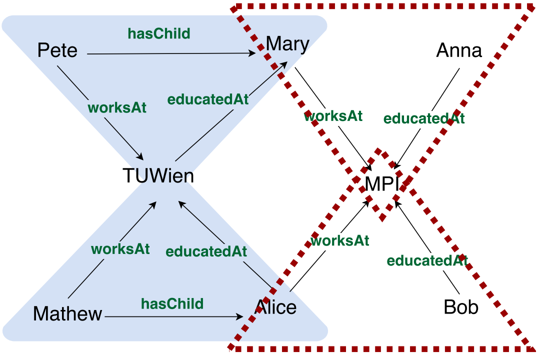
$$npi(r_1) = 1$$

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

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

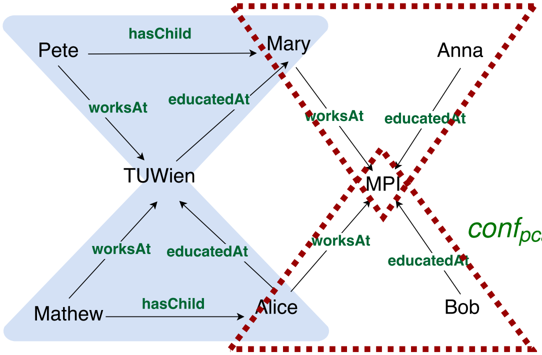
Completeness Confidence Example 2

$miss(hasChild, Alice) = 0$



Completeness Confidence Example 2

$miss(hasChild, Alice) = 0$



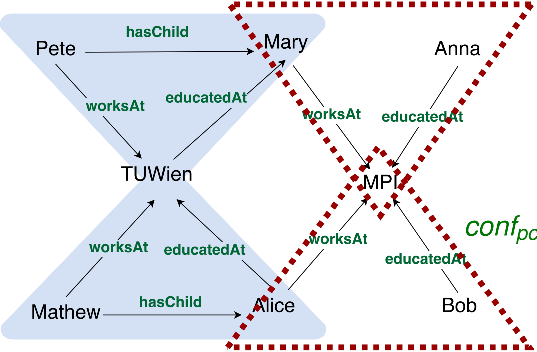
$$conf(r_2) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

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

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

Completeness Confidence Example 2

$miss(hasChild, Alice) = 0$



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

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

$$npi(r_2) = 0$$

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

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

Other Measures

precision$_{comp}$: penalize *r* that predict facts in complete areas

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

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

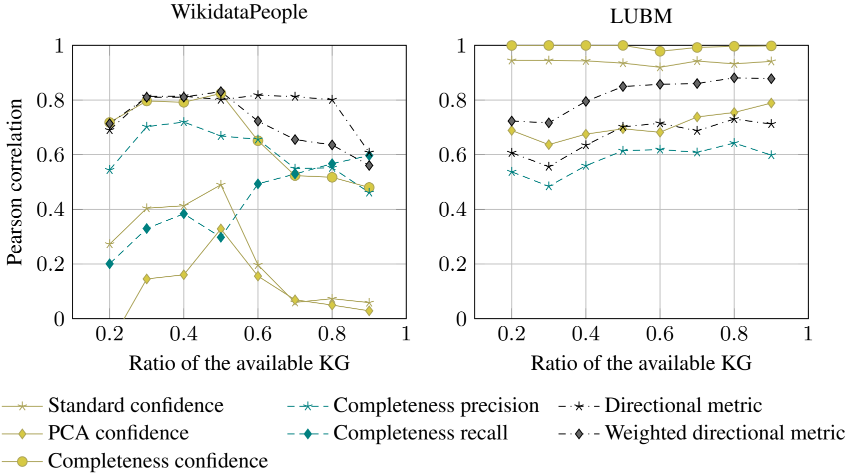
Creation of ideal KG:

- 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 $num(p, s)$
- 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 $num(p, s) = k$

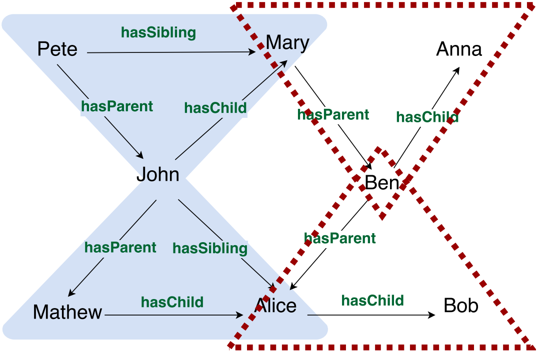
Completeness precision and recall

We define a precision and a recall:

$precision_{comp}(r) = 1 - \frac{npc(r)}{supp(\mathbf{B})}$ (ratio of ”complete” results)

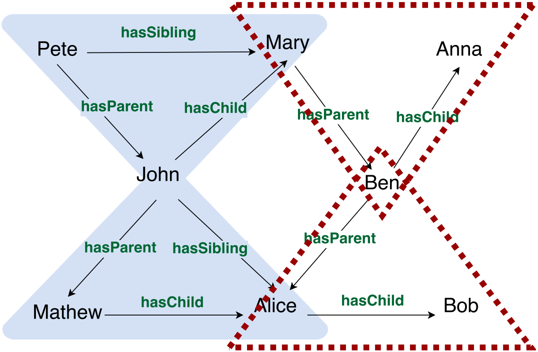
$recall_{comp}(r) = \frac{npi(r)}{\sum_s miss(h,s)}$ (ratio of ”incomplete” results filled)

Example with precision and recall



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

Example with precision and recall

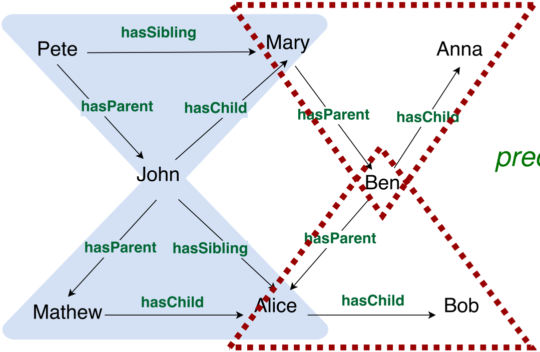


$npi(r) = 1 \; npc(r) = 0$

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

$miss(hasSibling, Mary) = 2$

Example with precision and recall



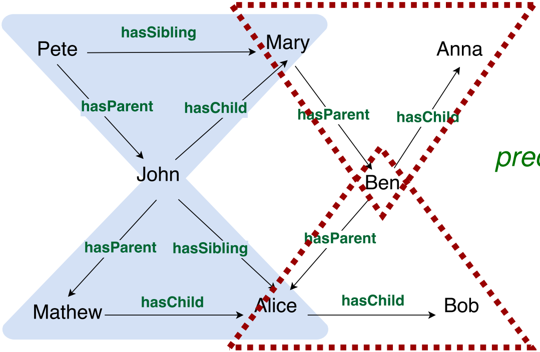
$npi(r) = 1$ $npc(r) = 0$

$precision_{comp}(r) = 1 - \frac{0}{4} = 1$

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

$miss(hasSibling, Mary) = 2$

Example with precision and recall



$$npi(r) = 1 \; npc(r) = 0$$
$$precision_{comp}(r) = 1 - \frac{0}{4} = 1$$
$$recall_{comp}(r) = \frac{1}{2}$$

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

Directional metric

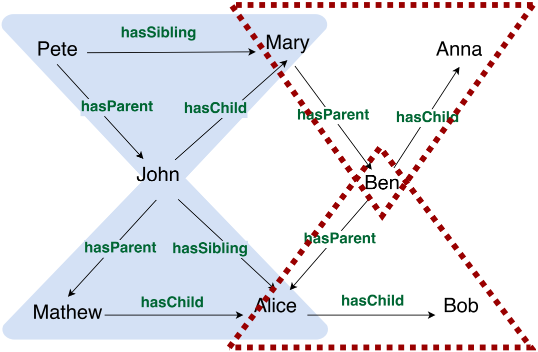
$$dir\_metric(r) = \frac{npi(r) - npc(r)}{2 \cdot (npi(r) + npc(r))} + 0.5$$

Directional metric

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

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

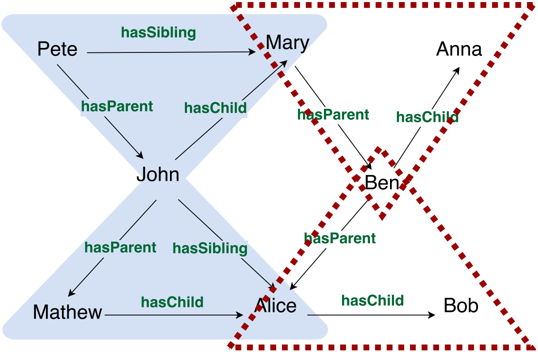
Example with directional metric



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

$$miss(hasSibling, Mary) = 2 \ \beta = 0.5$$

Example with directional metric

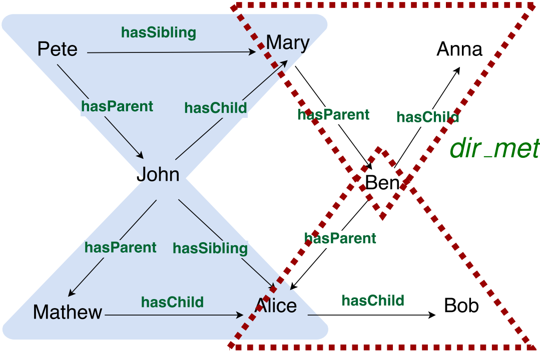


$npi(r) = 1 \; npc(r) = 0$

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

$miss(hasSibling, Mary) = 2 \; \beta = 0.5$

Example with directional metric



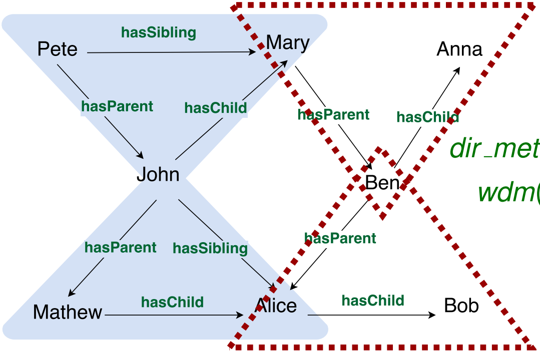
$$npi(r) = 1 \; npc(r) = 0$$

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

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

$$miss(hasSibling, Mary) = 2 \; \beta = 0.5$$

Example with directional metric



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

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

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

$\beta = 0.5$

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

$miss(hasSibling, Mary) = 2 \; \beta = 0.5$

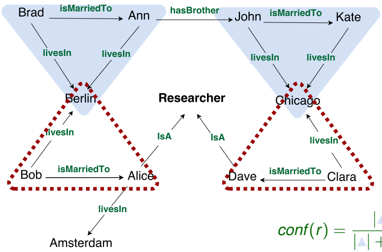
Knowledge Graph Completion

- **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



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

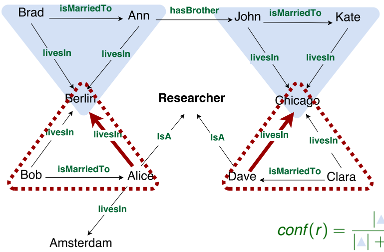
$r:\ livesIn(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z)$

AMIE [?], [?],
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



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

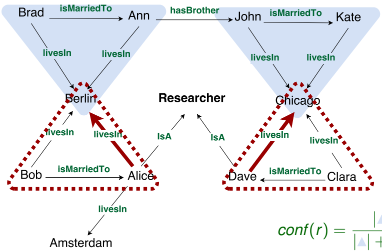
$r:\ livesIn(X, Z) \leftarrow isMarriedTo(Y, X), livesIn(Y, Z)$

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

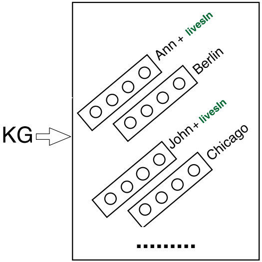


$$conf(r) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

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

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

Statistics-based approaches

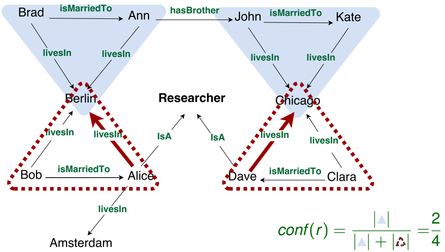


TransE [?], TEKE [?],
RESCAL [?], etc.

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

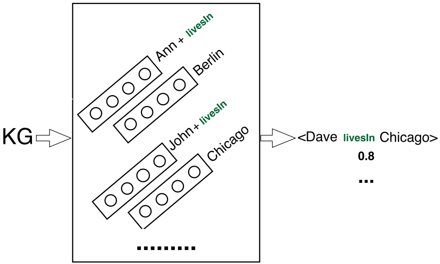


$$conf(r) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$$

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

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

Statistics-based approaches

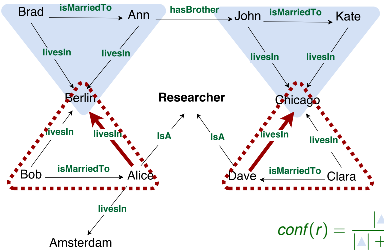


TransE [?], TEKE [?],
RESCAL [?], etc.

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

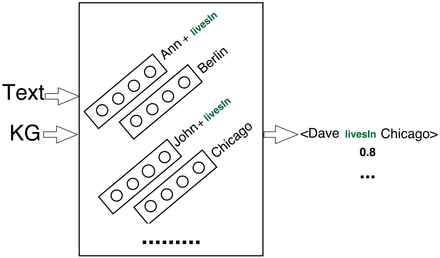


$conf(r) = \frac{|\blacktriangle|}{|\blacktriangle| + |\triangle|} = \frac{2}{4}$

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

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
  - 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

Rule Types

- **Horn:** AMIE [**?**]
  $livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z)$

Rule Types

- **Horn:** AMIE [**?**]
  $livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z)$
- **Nonmonotonic:** RUMIS [**?**], [**?**]
  $livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z), not\ researcher(Z)$

Rule Types

- **Horn:** AMIE [**?**]
  $livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z)$
- **Nonmonotonic:** RUMIS [**?**], [**?**]
  $livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z), not\ researcher(Z)$
- **Numerical:** CARL [**?**]
  $hasChild_{\geq 5}(X) \leftarrow hasFather(Y, X), hasSibling_{\geq 4}(Y)$

Rule Types

- **Horn:** AMIE [**?**]
  $livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z)$
- **Nonmonotonic:** RUMIS [**?**], [**?**]
  $livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z), not\ researcher(Z)$
- **Numerical:** CARL [**?**]
  $hasChild_{\geq 5}(X) \leftarrow hasFather(Y, X), hasSibling_{\geq 4}(Y)$
- **Disjunctive:**
  $male(Y) \vee female(Y) \leftarrow hasParent(X, Y)$

Rule Types

- **Horn:** AMIE [**?**]
  $livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z)$
- **Nonmonotonic:** RUMIS [**?**], [**?**]
  $livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z), not\ researcher(Z)$
- **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 [**?**]
  $livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z)$
- **Nonmonotonic:** RUMIS [**?**], [**?**]
  $livesIn(Z, Y) \leftarrow livesIn(X, Y), marriedTo(X, Z), not\ researcher(Z)$
- **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:**
  $\exists Y\ hasParent(X, Y) \leftarrow person(X)$
- **Temporal constraints:**
  $\perp \leftarrow bornIn(X, Y), after(Y, Z), studied(X, Z)$

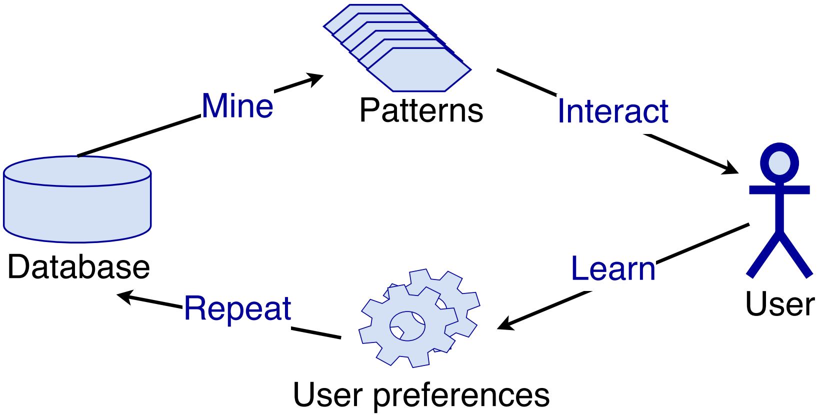
Related Works

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

- **Rule mining with external support**
  - Interactive pattern mining [**?**],
    [**?**]
  - Interactive association rule mining [**?**]

Mine-Interact-Learn-Repeat

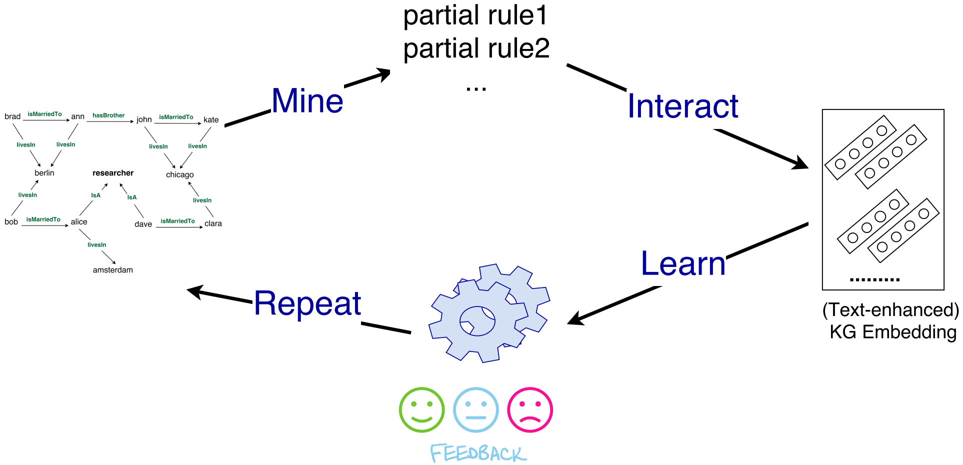
Mimic "mine-interact-learn-repeat" schema [?]



Mine-Interact-Learn-Repeat

Mimic "mine-interact-learn-repeat" schema [?]

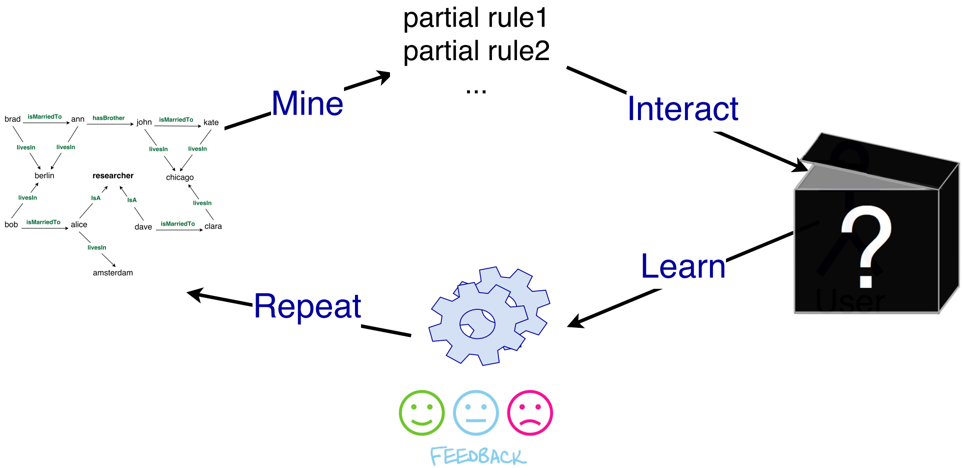
Establish "user-in-the-loop" inspired interaction between the rule mining algorithm and the embedding model



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

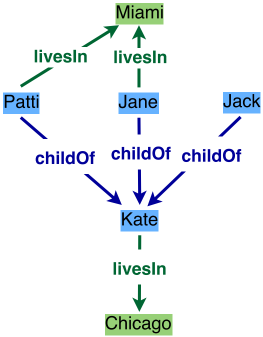
Q1 **(Interact)** What kind of feedback is required/possible to obtain from the "black box" to organize convenient and effective interaction process?

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?

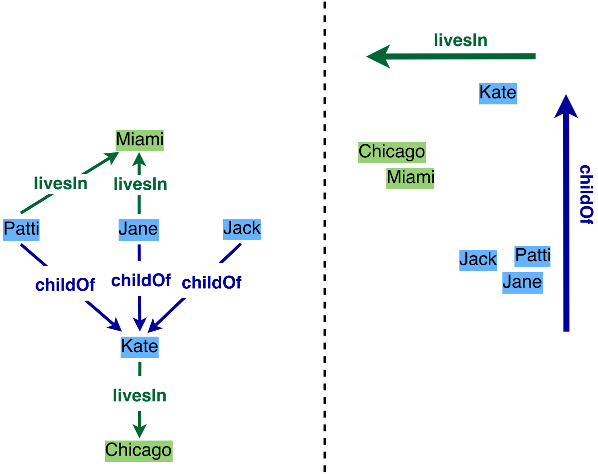
Embedding-based Methods

- **Intuition:** For $\langle s, p, o \rangle$ in KG, find $\mathbf{s}, \mathbf{p}, \mathbf{o}$ 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



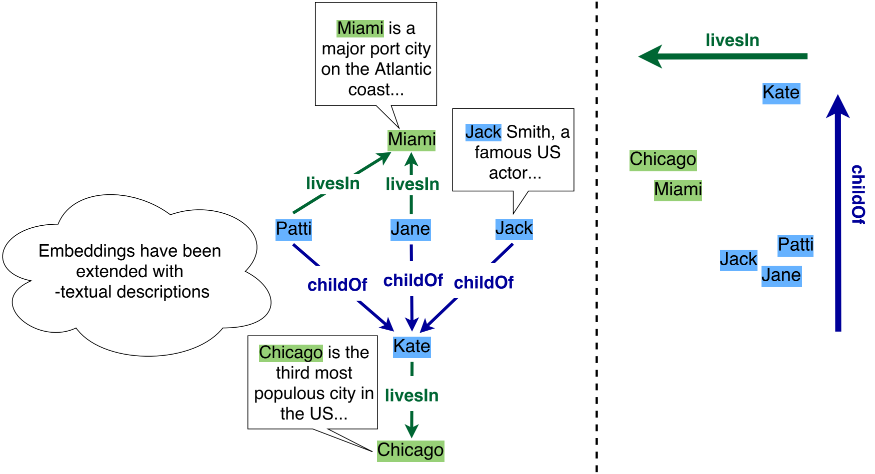
Embedding-based Methods

- **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



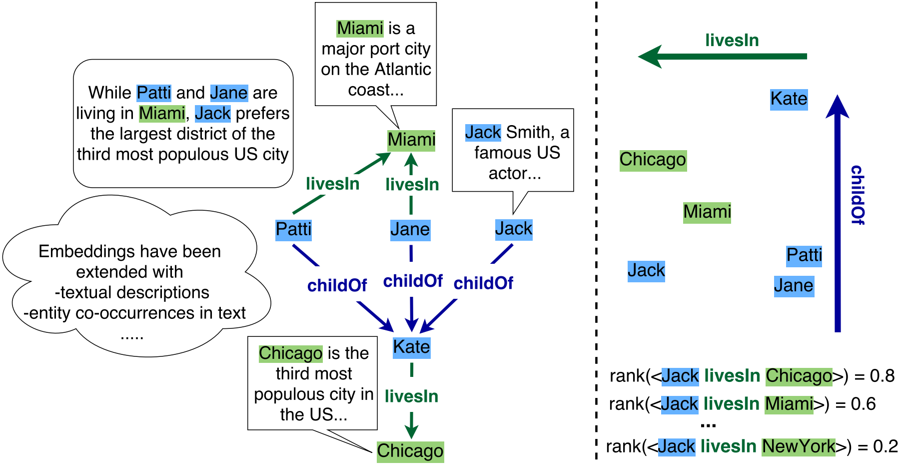
Embedding-based Methods

- **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)

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

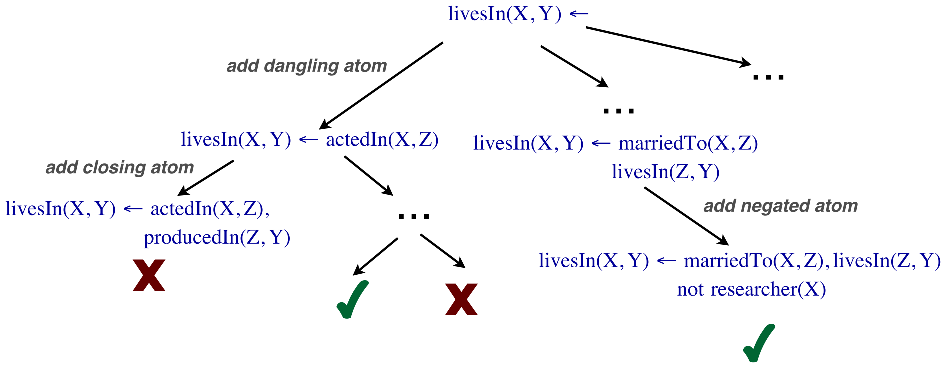
  2. extend the queue by applying refinement operators

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

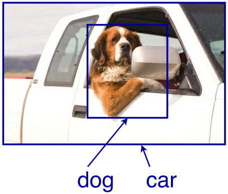
Q1 **(Interact)** What kind of feedback is required/possible to obtain from the "black box" to organize convenient and effective interaction process?

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?

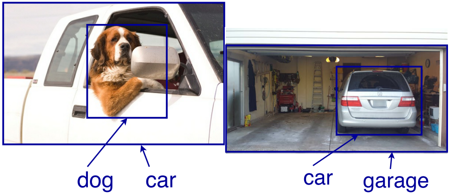
- Ideally, we want to learn the structure of most promising rules, i.e., the best rules have at most 5 atoms, 4 variables, etc..

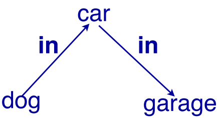
Commonsense Rule Mining from Hybrid Sources



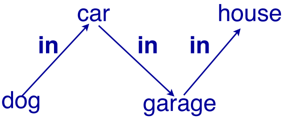
$\text{dog} \xrightarrow{\textbf{in}} \text{car}$

Commonsense Rule Mining from Hybrid Sources

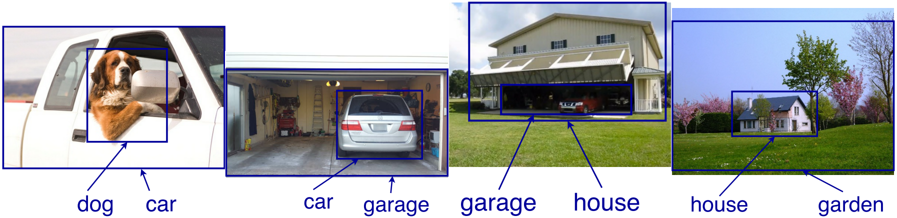


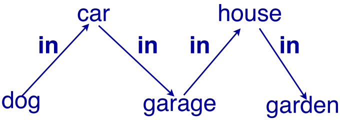


Commonsense Rule Mining from Hybrid Sources

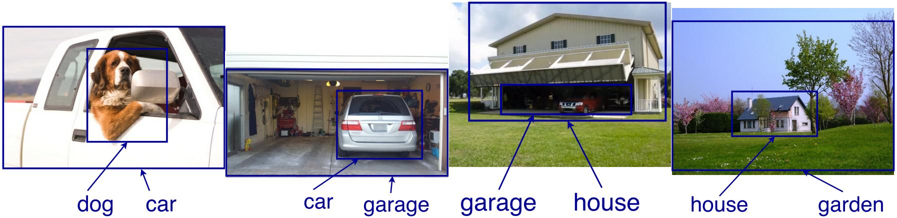


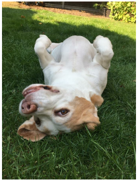
Commonsense Rule Mining from Hybrid Sources



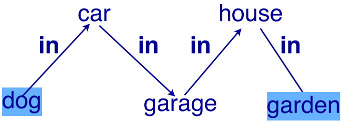


Commonsense Rule Mining from Hybrid Sources

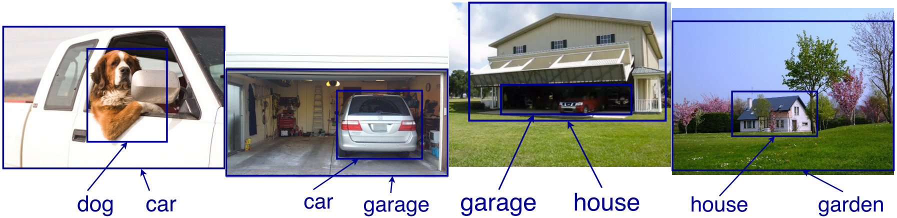


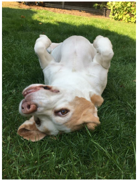


Dog enjoying the sun in the garden

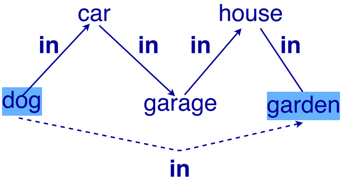


Commonsense Rule Mining from Hybrid Sources

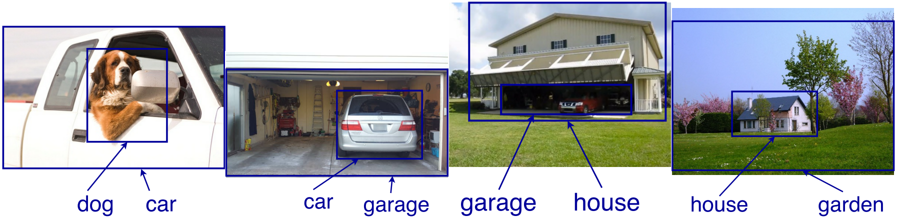


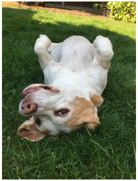


Dog enjoying the sun in the garden



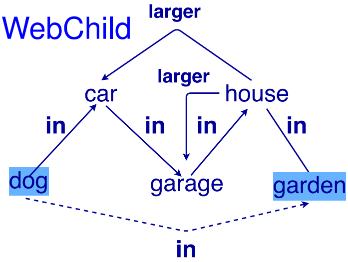
Commonsense Rule Mining from Hybrid Sources



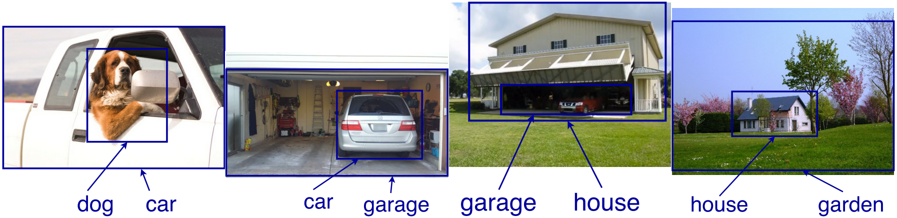


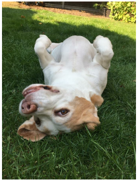
Dog enjoying the sun in the garden

+WebChild

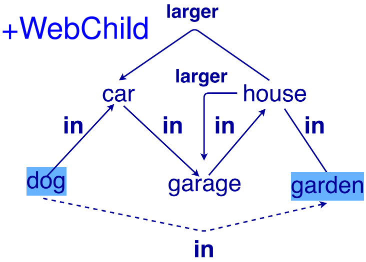


Commonsense Rule Mining from Hybrid Sources





Dog enjoying the sun in the garden



Desired output:

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

$\dots$