Predicting Completeness in Knowledge Bases

Luis Galárraga, Simon Razniewski, Antoine Amarilli, Fabian Suchanek

February 8th, 2017

WSDM'17, Cambridge

Predicting completeness in KBs

- KBs are highly incomplete.

Predicting completeness in KBs

- KBs are highly incomplete.
  - 2% of people have a father in Wikidata.

Predicting completeness in KBs

- KBs are highly incomplete.
  - 2% of people have a father in Wikidata.
- We do not know where the incompleteness lies.

Predicting completeness in KBs

- KBs are highly incomplete.
  - 2% of people have a father in Wikidata.
- We do not know where the incompleteness lies.
  - A person without spouse in the KB could be incomplete or single.

Predicting completeness in KBs

- KBs are highly incomplete.
  - 2% of people have a father in Wikidata.
- We do not know where the incompleteness lies.
  - A person without spouse in the KB could be incomplete or single.
- Problems for data producers and consumers.

Predicting completeness in KBs

- KBs are highly incomplete.
  - 2% of people have a father in Wikidata.
- We do not know where the incompleteness lies.
  - A person without spouse in the KB could be incomplete or single.
- Problems for data producers and consumers.
  - Consumers: no completeness guarantees for queries.
  - Producers: which parts of the KB need to be populated?

Completeness

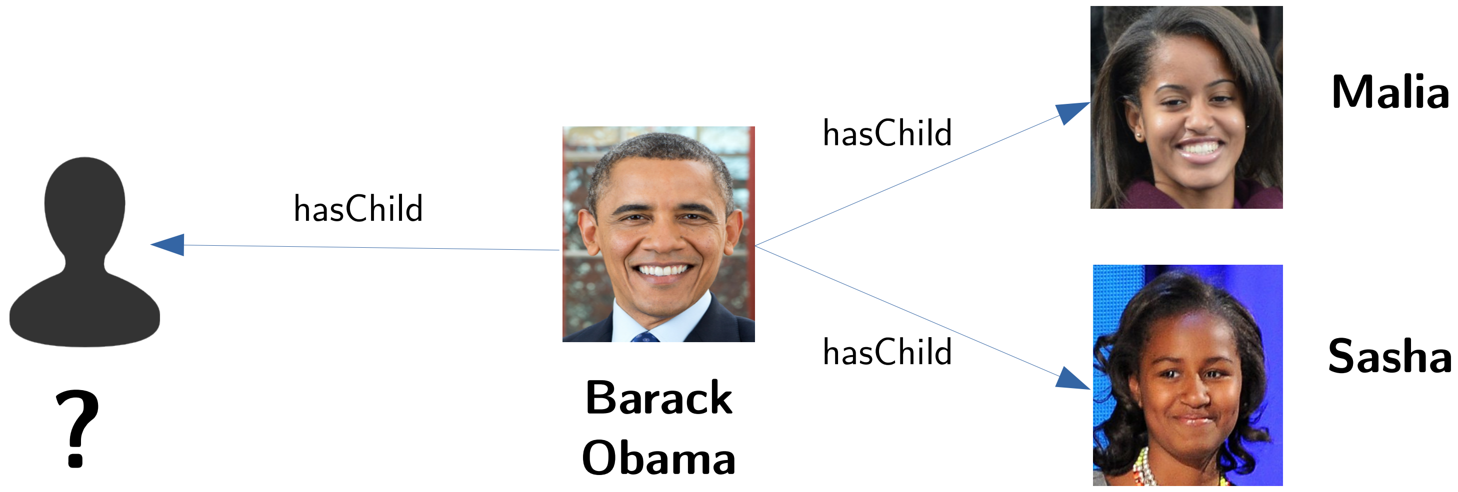
We focus on queries of the form

SELECT ?x WHERE { subject relation ?x }

Completeness

We focus on queries of the form

SELECT ?x WHERE { Barack Obama hasChild ?x }



Completeness

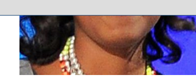
We focus on queries of the form

SELECT ?x WHERE { Barack Obama hasChild ?x }

Goal: Study different signals to predict if a query of the form {o : r(s, o) } is complete in a KB.

?

Obama



Completeness oracles

- Function that assigns a completeness value to pairs subject-relation (s, r).

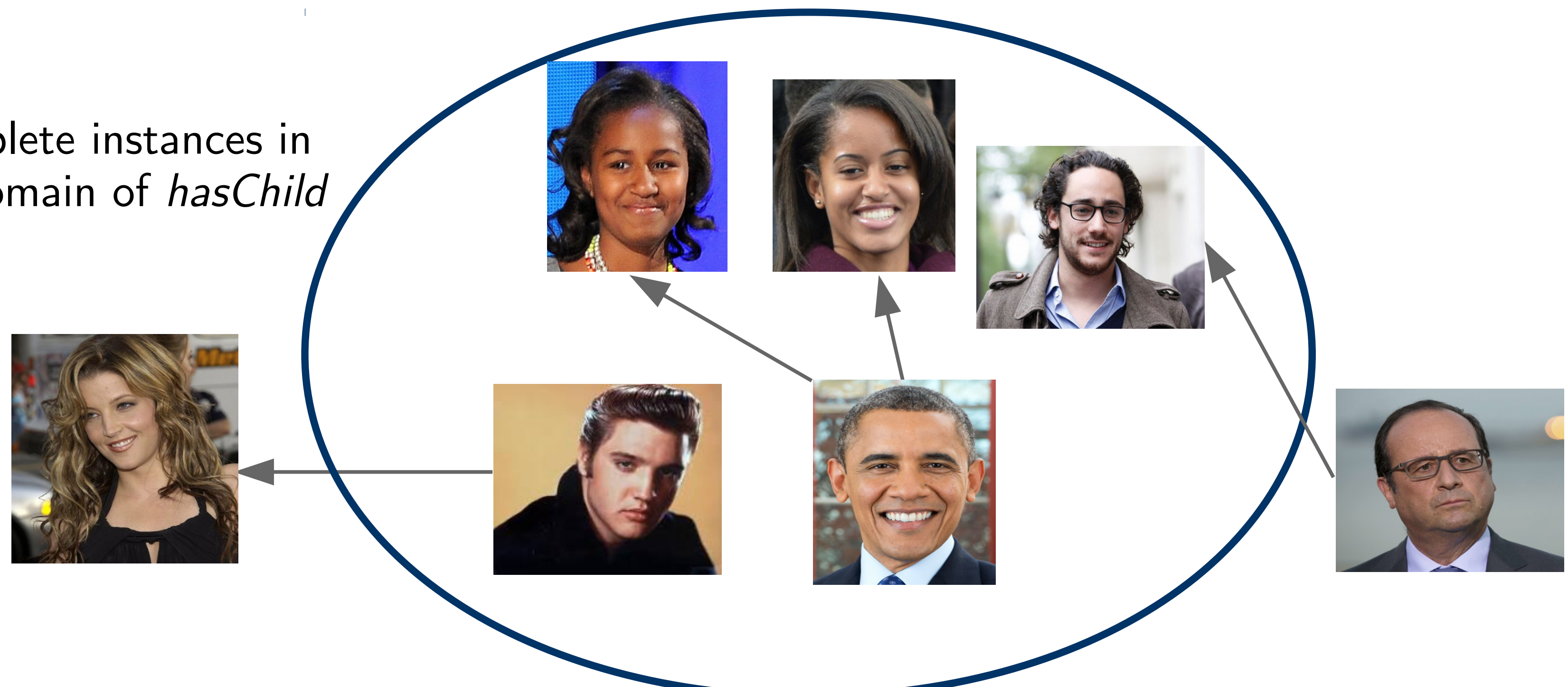
Completeness oracles

- Function that assigns a completeness value to pairs subject-relation (s, r).
  - PCA oracle: (s, r) is **complete** if the KB knows at least one object *o*.

Completeness oracles

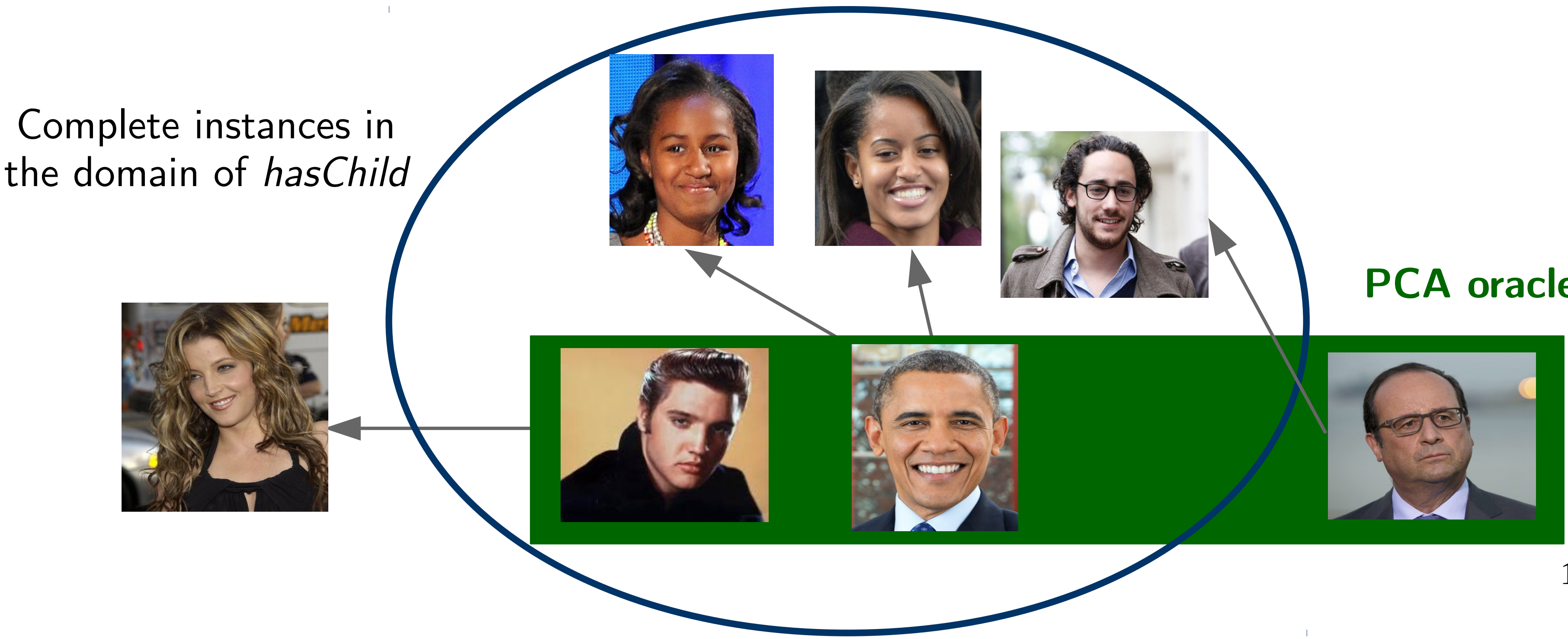
- Function that assigns a completeness value to pairs subject-relation (s, r).
  - <u>PCA oracle</u>: (s, r) is **complete** if the KB knows at least one object *o*.

Complete instances in the domain of *hasChild*



Completeness oracles

- Function that assigns a completeness value to pairs subject-relation (s, r).
  - <u>PCA oracle</u>: (s, r) is **complete** if the KB knows at least one object *o*.



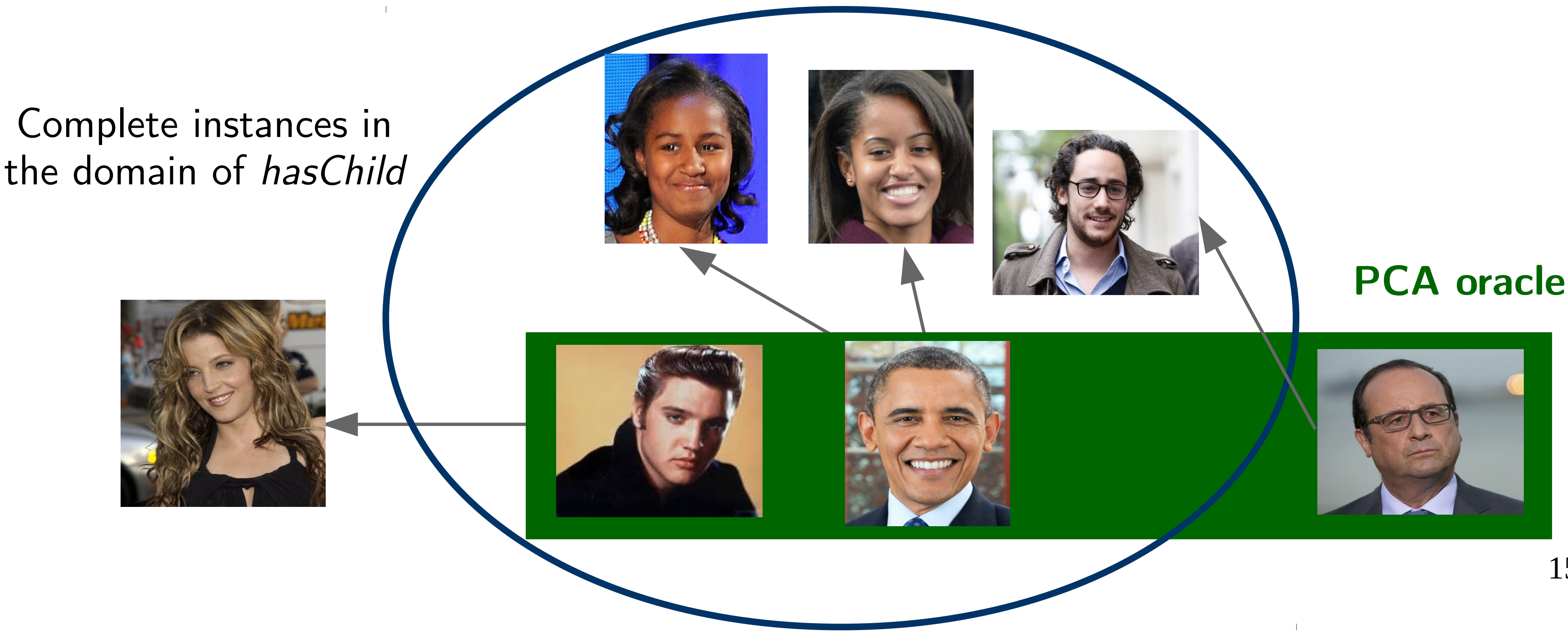
Completeness oracles

Oracles can be evaluated via precision and recall.

PCA oracle

Precision = 2/3

Recall = 2/5



Completeness oracles

- CWA: $cwa(s, r) = true$
- PCA: $pca(s, r) = \exists\ o : r(s, o)$
- Cardinality: $card(s, r) = \#(o : r(s, o)) \geq k$
- Popular entities: $popularity_{pop}(s, r) = pop(s)$
- No-chg over time: $nochange_{chg}(s, r) = \sim chg(s, r)$
- Star : $star_{r_1,..,r_n}(s, r) = \forall\ i \in \{1,..,n\} : \exists\ o : r_i(s, o)$
- Class: $class_c(s, r) = type(s, c)$
- AMIE (rule mining)

Completeness oracles

- CWA: $cwa(s, r) = true$
- PCA: $pca(s, r) = \exists\ o : r(s, o)$
- Cardinality: $card(s, r) = \#(o : r(s, o)) \geq k$
- Popular entities: $popularity_{pop}(s, r) = pop(s)$
- No-chg over time: $nochange_{chg}(s, r) = {\sim}chg(s, r)$
- Star : $star_{r_1,..,r_n}(s, r) = \forall\ i \in \{1,..,n\} : \exists\ o : r_i(s, o)$
- Class: $class_c(s, r) = type(s, c)$
- AMIE (rule mining)

Learned oracles

AMIE oracle

- It combines all the other oracles using rules.

AMIE oracle

- It combines all the other oracles using rules.

notype(x, Adult), type(x, Person) ⇒ complete(x, hasChild) **$\text{class}_{\text{non-adult}}$(s, r)**

dateOfDeath(x, y), lessThan_1(x, placeOfDeath) ⇒ incomplete(x, placeOfDeath)

AMIE oracle

- It combines all the other oracles using rules.

notype(x, Adult), type(x, Person) ⇒ complete(x, hasChild) **$\text{class}_{\text{non-adult}}$(s, r)**

dateOfDeath(x, y), lessThan_1(x, placeOfDeath) ⇒ incomplete(x, placeOfDeath)

- Training data obtained by two means:
  - Automatic: e.g., everyone must have a nationality.
  - Crowd-sourcing: ask mechanical turks for more objects in the web.

Experimental evaluation

Evaluating oracles

F1 measure of the oracles in YAGO3.

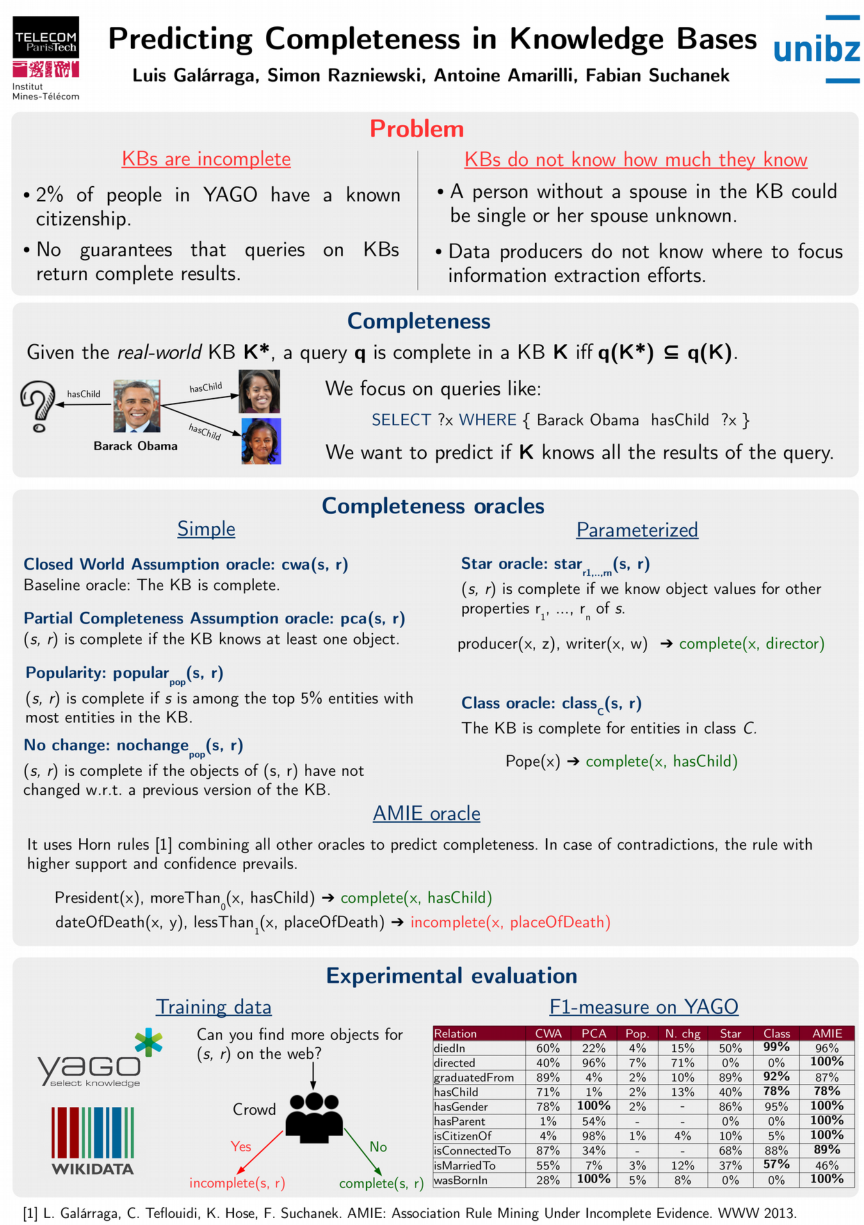
Relation	CWA	PCA	Class	AMIE
diedIn	60%	22%	**99%**	96%
directed	40%	96%	0%	**100%**
graduatedFrom	89%	4%	**92%**	87%
hasChild	71%	1%	**78%**	**78%**
hasGender	78%	**100%**	95%	**100%**
hasParent	1%	54%	0%	**100%**
isCitizenOf	4%	98%	5%	**100%**
isConnectedTo	87%	34%	88%	**89%**
isMarriedTo	55%	7%	**57%**	46%
wasBornIn	28%	**100%**	0%	**100%**

Summary

- It is possible to predict completeness in KBs with 100% precision in some cases.
  - By combining different simple oracles (signals).
- Future work
  - Study of more signals of completeness
  - Reasoning with completeness information
  - Completeness predictions as counter-evidence for learning methods in KBs.

More information

Visit my poster ->



Predicting Completeness in Knowledge Bases

Luis Galárraga, Simon Razniewski, Antoine Amarilli, Fabian Suchanek

TELECOM ParisTech, Institut Mines-Télécom, unibz

Problem

KBs are incomplete

- 2% of people in YAGO have a known citizenship.
- No guarantees that queries on KBs return complete results.

KBs do not know how much they know

- A person without a spouse in the KB could be single or her spouse unknown.
- Data producers do not know where to focus information extraction efforts.

Completeness

Given the *real-world* KB **K***, a query **q** is complete in a KB **K** iff **q(K*) ⊆ q(K)**.

We focus on queries like:

SELECT ?x WHERE { Barack Obama hasChild ?x }

We want to predict if **K** knows all the results of the query.

Completeness oracles

Simple

Closed World Assumption oracle: cwa(s, r)
Baseline oracle: The KB is complete.

Partial Completeness Assumption oracle: pca(s, r)
(*s*, *r*) is complete if the KB knows at least one object.

Popularity: $popular_{pop}$(s, r)
(*s*, *r*) is complete if *s* is among the top 5% entities with most entities in the KB.

No change: $nochange_{pop}$(s, r)
(*s*, *r*) is complete if the objects of (s, r) have not changed w.r.t. a previous version of the KB.

Parameterized

Star oracle: $star_{r1,...,m}$(s, r)
(*s*, *r*) is complete if we know object values for other properties r_1, ..., r_n of *s*.

producer(x, z), writer(x, w) ➔ complete(x, director)

Class oracle: $class_C$(s, r)
The KB is complete for entities in class *C*.

Pope(x) ➔ complete(x, hasChild)

AMIE oracle

It uses Horn rules [1] combining all other oracles to predict completeness. In case of contradictions, the rule with higher support and confidence prevails.

President(x), $moreThan_0$(x, hasChild) ➔ complete(x, hasChild)
dateOfDeath(x, y), $lessThan_1$(x, placeOfDeath) ➔ incomplete(x, placeOfDeath)

Experimental evaluation

Training data

F1-measure on YAGO

Relation	CWA	PCA	Pop.	N. chg	Star	Class	AMIE
diedIn	60%	22%	4%	15%	50%	**99%**	96%
directed	40%	96%	7%	71%	0%	0%	**100%**
graduatedFrom	89%	4%	2%	10%	89%	**92%**	87%
hasChild	71%	1%	2%	13%	40%	**78%**	**78%**
hasGender	78%	**100%**	2%	-	86%	95%	**100%**
hasParent	1%	54%	-	-	0%	0%	**100%**
isCitizenOf	4%	98%	1%	4%	10%	5%	**100%**
isConnectedTo	87%	34%	-	-	68%	88%	**89%**
isMarriedTo	55%	7%	3%	12%	37%	**57%**	46%
wasBornIn	28%	**100%**	5%	8%	0%	0%	**100%**

[1] L. Galárraga, C. Teflouidi, K. Hose, F. Suchanek. AMIE: Association Rule Mining Under Incomplete Evidence. WWW 2013.