Exception-enriched Rule Learning from Knowledge Graphs

Mohamed Gad-Elrab[1], *Daria Stepanova[1], Jacopo Urbani [2], Gerhard Weikum[1]*

[1]Max-Planck-Institut für Informatik, Saarland Informatics Campus, Germany

[2] Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

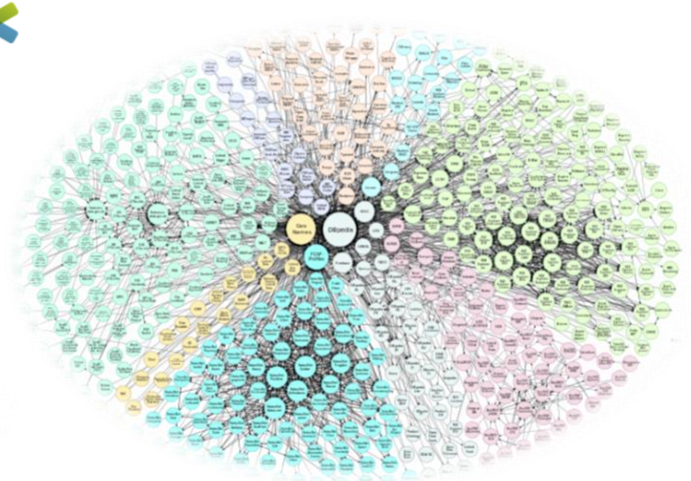
21st October 2016

Knowledge Graphs (KGs)

- Huge collection of $< subject, predicate, object >$ triples
- Positive facts under Open World Assumption (OWA)
- Possibly incomplete and/or inaccurate

yago select knowledge

WIKIDATA

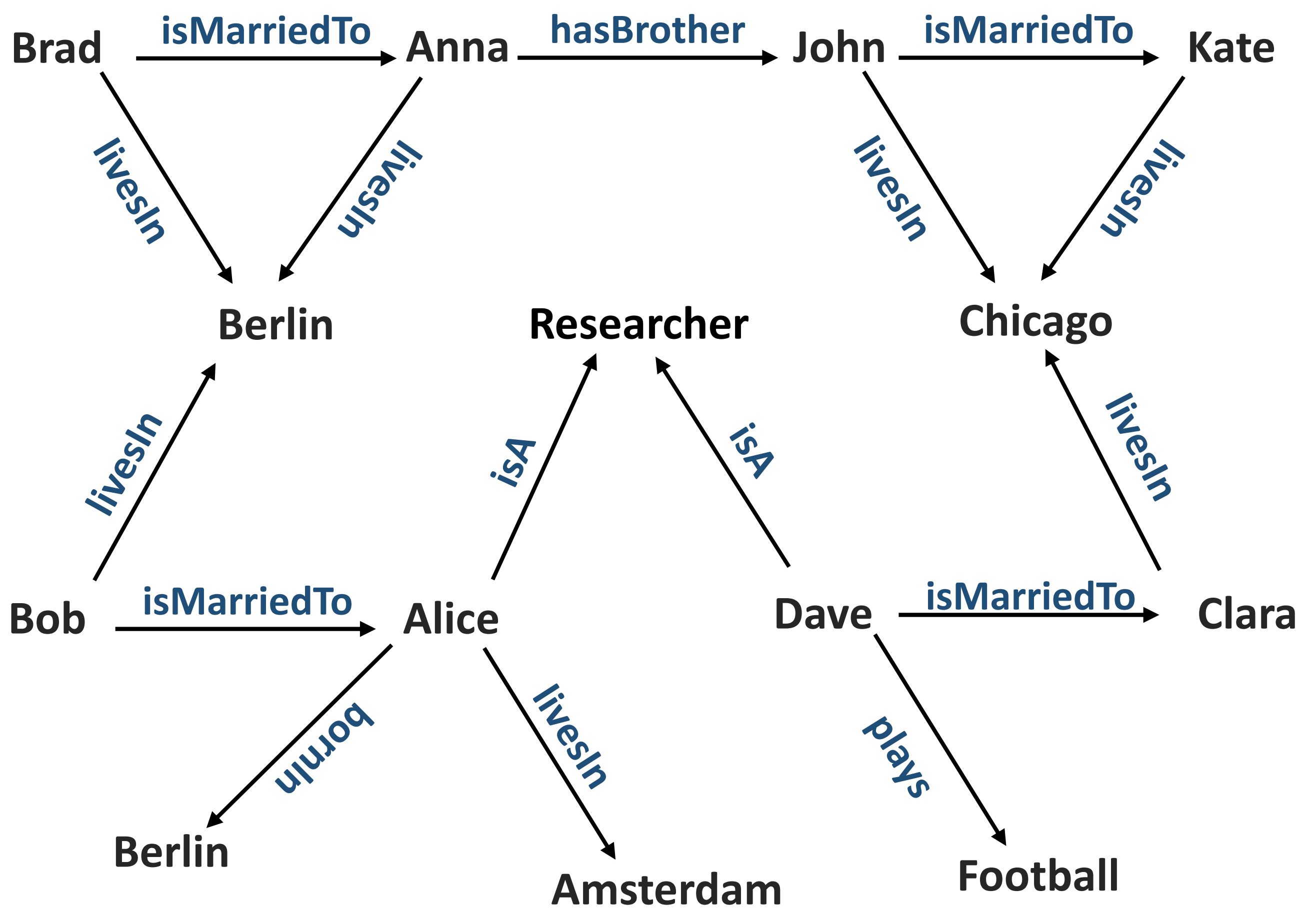


DBpedia

BabelNet

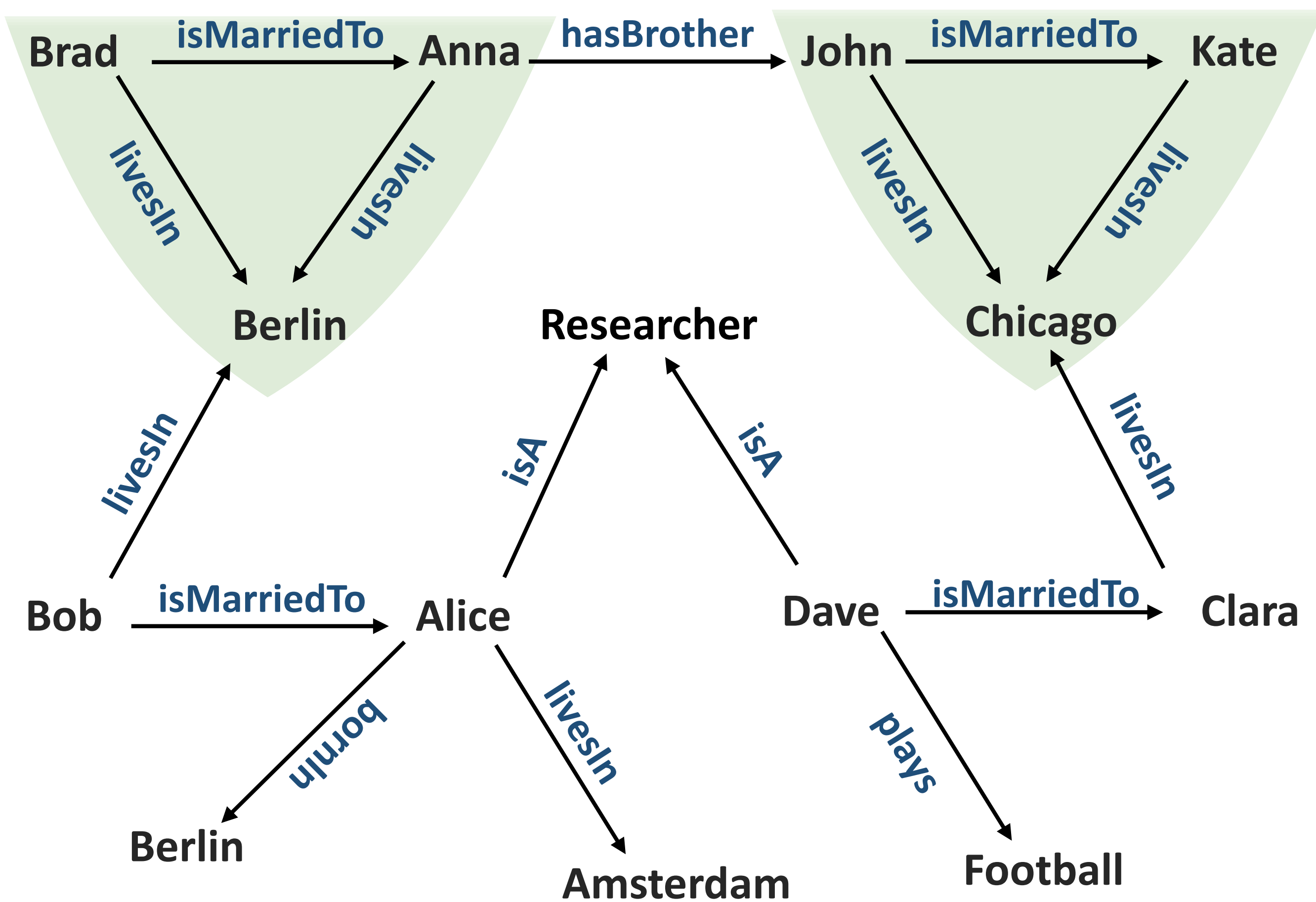
Freebase

Mining Rules from KGs



Mining Rules from KGs

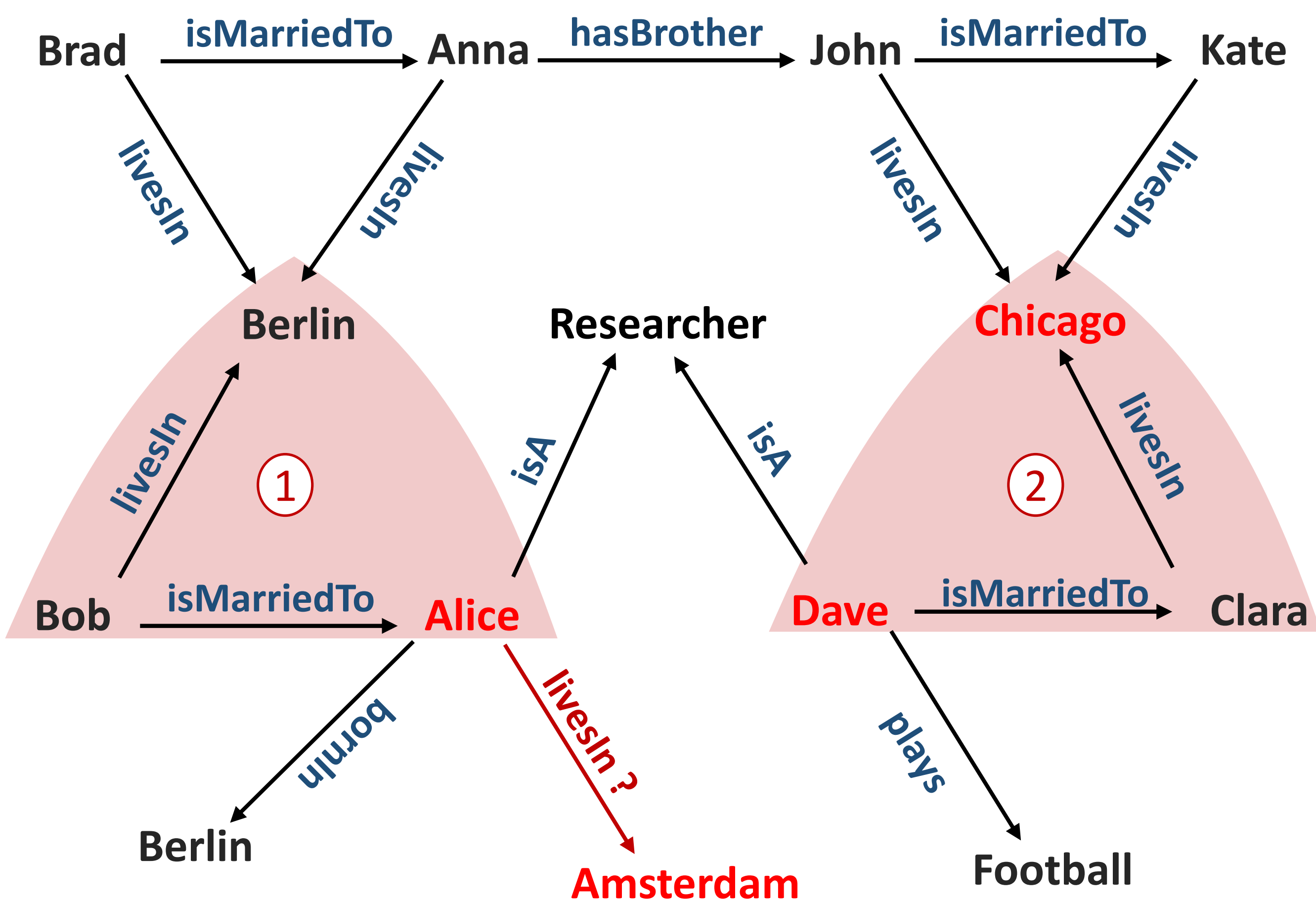
$$r: livesIn(X,Z) \leftarrow isMarriedTo(X,Y), livesIn(Y,Z)$$



[Galárraga et al., 2015]

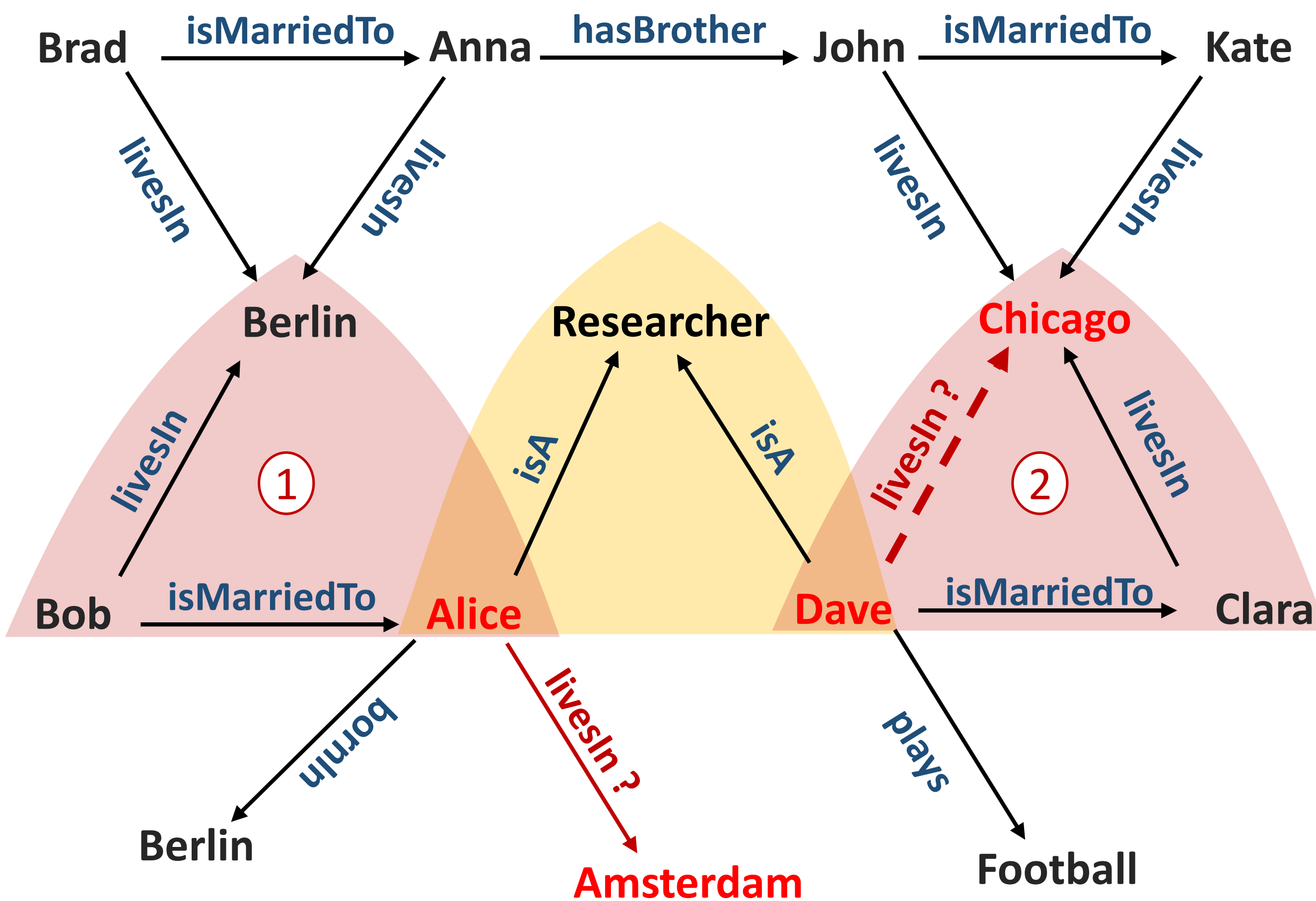
Mining Rules from KGs

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



Our Goal

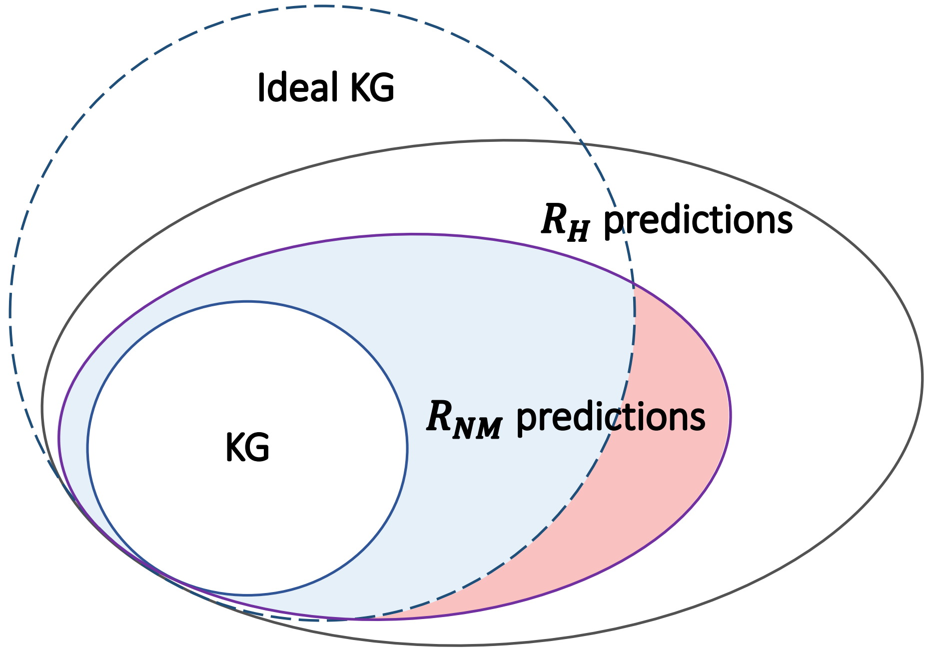
$$r: livesIn(X,Z) \leftarrow isMarriedTo(X,Y), livesIn(Y,Z), not\ isA(X,res)$$



Problem Statement

- Quality-based theory revision problem

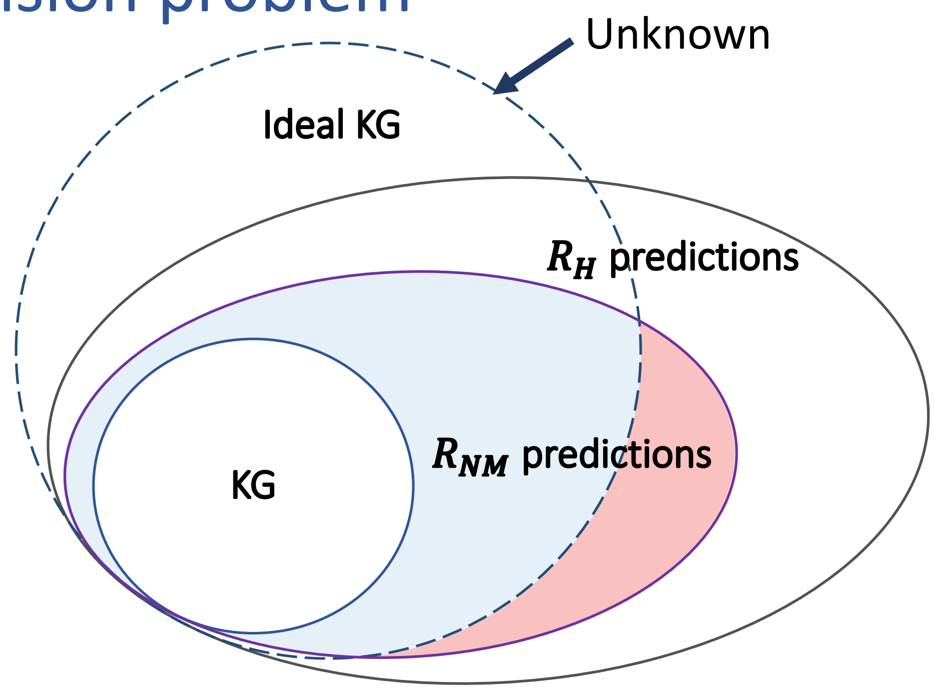
- Given
  - Knowledge graph KG
  - Set of Horn rules R_H



- Find the nonmonotonic revision R_{NM} of R_H

Problem Statement

- Quality-based theory revision problem



- Given
  - Knowledge graph KG
  - Set of Horn rules R_H

- Find the nonmonotonic revision R_{NM} of R_H
  - Maximize *top-k avg. confidence*
  - Minimize *conflicting prediction*

Problem Statement: Conflicting Predictions

- Defining conflicts

$$R = \begin{cases} r_1: releasedInJP(X) \leftarrow isGame(X), isBasedOnJPAnime\,(X) \\ r_2: releasedInJP(X) \leftarrow hasJPComposer(X), not\ publisherUSA(X) \end{cases}$$

Problem Statement: Conflicting Predictions

- Defining conflicts

$$R = \begin{cases} r_1 : releasedInJP(X) \leftarrow isGame(X), isBasedOnJPAnime\ (X) \\ r_2 : releasedInJP(X) \leftarrow hasJPComposer(X), not\ publisherUSA(X) \end{cases}$$

$$\{isGame(a), isBasedOnJPAnime(a), hasJPComposer(a), publisherUSA(a)\}$$

$a =$

Problem Statement: Conflicting Predictions

- Defining conflicts

$$R = \begin{cases} r_1: releasedInJP(X) \leftarrow isGame(X), isBasedOnJPAnime\ (X) \\ r_2: releasedInJP(X) \leftarrow hasJPComposer(X), not\ publisherUSA(X) \end{cases}$$

$$\{isGame(a), isBasedOnJPAnime(a), hasJPComposer(a), publisherUSA(a)\}$$

$a =$

- Measuring conflicts (auxiliary rules)

$$r_2: releasedInJP(X) \leftarrow hasJPComposer(X), not\ publisherUSA(X)$$
$$r_2^{aux}: not\_releasedInJP(X) \leftarrow hasJPComposer(X), publisherUSA(X)$$

$$(releasedInJP(a), not\_releasedInJP(a))$$

Approach Overview

Step 1
- Mining Horn Rules

Step 2
- Extracting Exception Witness Set (EWS)

Step 3
- Constructing Candidate Revisions

Step 4
- Selecting the Best Revision

Step 1: Mining Horn Rules

$$r_i: livesInUSA(X) \leftarrow bornInUSA(X)$$

		bornInUSA	livesInUSA	stateless	emigrant	singer	poet
Normal	p1	✓	✓				
	p2	✓	✓				
	p3	✓	✓				
	p4	✓	✓			✓	
	p5	✓	✓	✓			
Ab-normal	p6	✓		✓			
	p7	✓		✓			
	p8	✓		✓	✓		✓
	p9	✓			✓		✓
	p10	✓			✓	✓	✓
	p11	✓				✓	✓

Step 2: Extracting Exception Witness Set (EWS)

$$r_i: livesInUSA(X) \leftarrow bornInUSA(X)$$

		bornInUSA	livesInUSA	stateless	emigrant	singer	poet
Normal	p1	✓	✓				
	p2	✓	✓				
	p3	✓	✓				
	p4	✓	✓			✓	
	p5	✓	✓	✓			
Ab-normal	p6	✓		✓			
	p7	✓		✓			
	p8	✓		✓	✓		✓
	p9	✓			✓		✓
	p10	✓			✓	✓	✓
	p11	✓				✓	✓

$$EWS_i = \{emigrant(X), poet(X)\}$$

Step 3: Constructing Candidate Revisions

- Horn rules

$$R = \begin{cases} r_1\colon livesInUSA(X) \leftarrow bornInUSA(X) & EWS = \{poet(X), emigrant(X), \ldots\} \\ r_2\colon emigrant(X) \leftarrow stateless(X) & EWS = \{e_1(X), e_2(X), \ldots\} \end{cases}$$

- Rule revisions

$r_1\colon livesInUSA(X) \leftarrow bornInUSA(X), not\ poet(X)$
$r_1\colon livesInUSA(X) \leftarrow bornInUSA(X), not\ emigrant(X)$
…

Step 4: Selecting the Best Revision

Finding globally best revision is expensive!

- Naïve ranker
  - For each rule, pick the revision that maximizes confidence
  - Works in isolation from other rules
- Partial materialization ranker
  - KGs are incomplete!
  - Augment the original KG with predictions of other rules
  - Rank revisions on avg. confidence of the rule and its auxiliary.

Ranking Rule's Revisions

- Partial materialization

$$r_i: livesInUSA(X) \leftarrow bornInUSA(X)$$

		bornInUSA	livesInUSA	stateless	emigrant	singer	poet
Normal	p1	✓	✓				
	p2	✓	✓				
	p3	✓	✓				
	p4	✓	✓			✓	
	p5	✓	✓	✓			
Ab-normal	p6	✓		✓			
	p7	✓		✓			
	p8	✓		✓	✓		✓
	p9	✓			✓		✓
	p10	✓			✓	✓	✓
	p11	✓				✓	✓

Ranking Rule's Revisions

- Partial materialization

$$r_i: \boldsymbol{livesInUSA}(X) \leftarrow \boldsymbol{bornInUSA}(X)$$

		bornInUSA	livesInUSA	stateless	emigrant	singer	poet
Normal	p1	✓	✓				
	p2	✓	✓				
	p3	✓	✓				
	p4	✓	✓			✓	✓
	p5	✓	✓	✓		✓	
Ab-normal	p6	✓	✓	✓			
	p7	✓		✓	✓		
	p8	✓		✓	✓		✓
	p9	✓	✓		✓	✓	✓
	p10	✓			✓	✓	✓
	p11	✓			✓	✓	✓

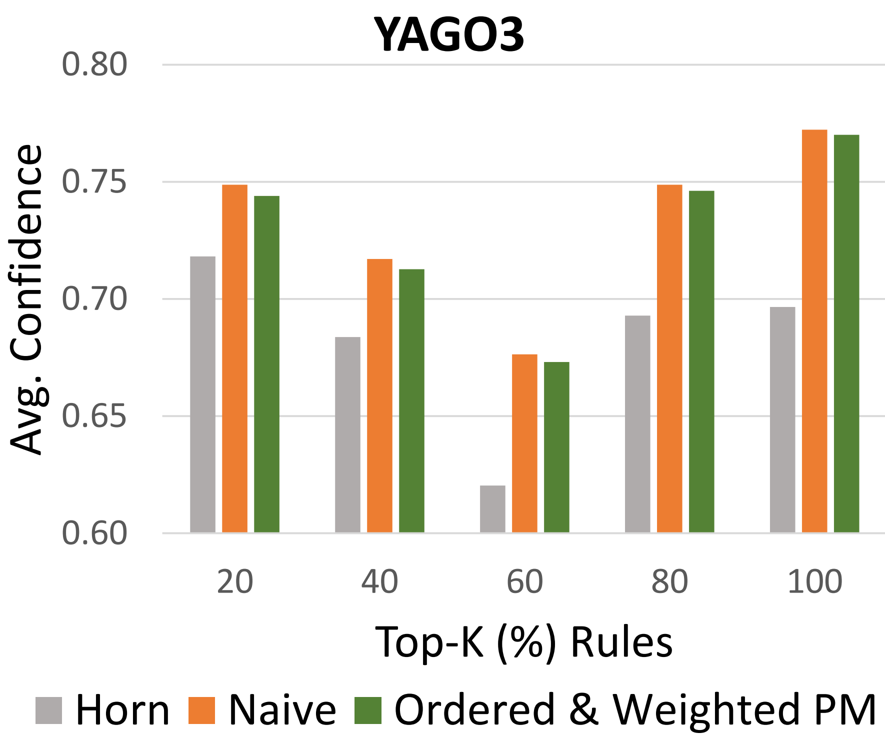
Ranking Rule's Revisions

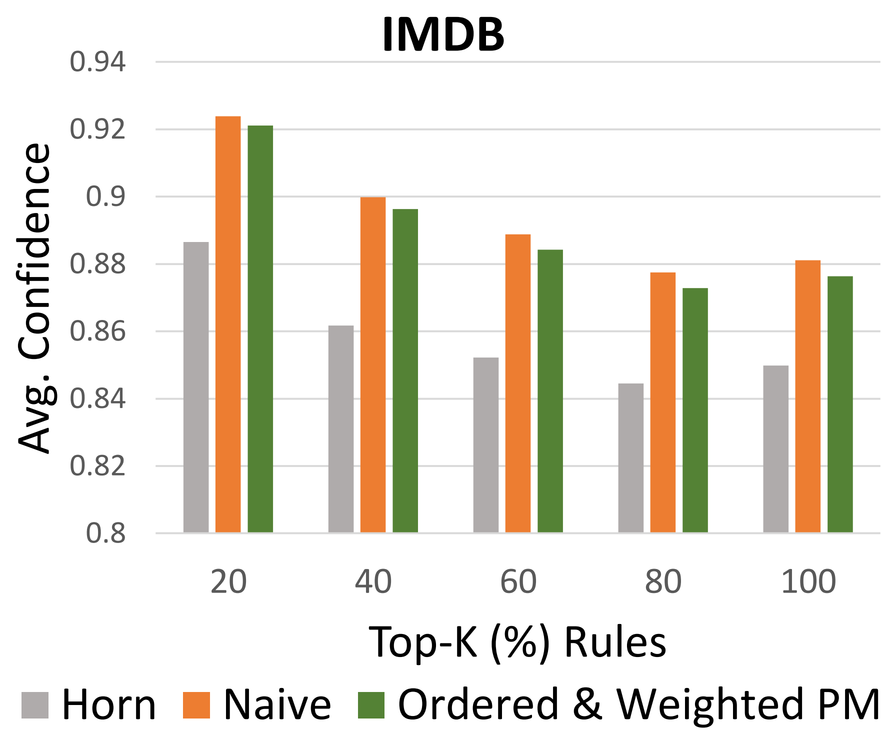
- Ordered partial materialization ranker
  - Only rules with higher quality
- Ordered weighted partial materialization ranker
  - KG fact weight = 1
  - Predicted facts inherit their weights from the rules

Experiments

- Ruleset quality

	yago	IMDb
Facts	**10M**	**2M**
Rules	**10K**	**25K**

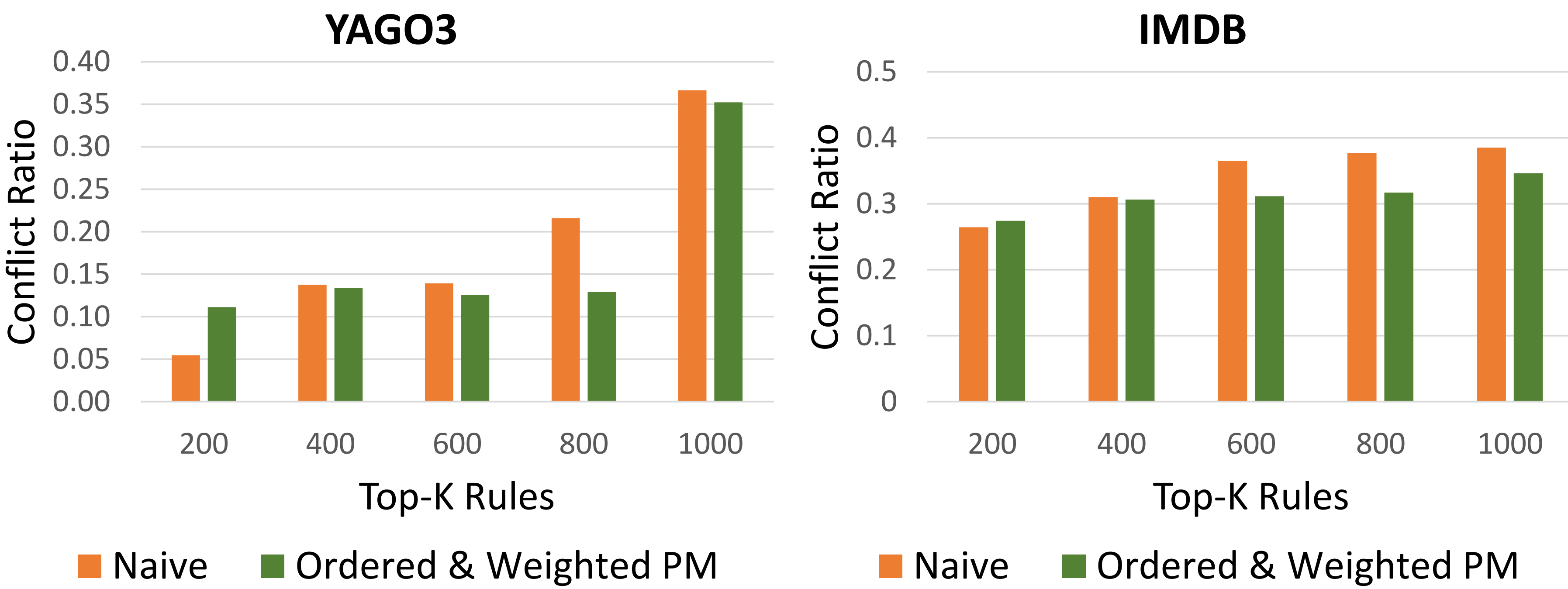




*Higher is better

Experiments

- Predictions consistency



*Lower is better

Experiments

- Examples

$$isMountain(X) \leftarrow isInAustria(X), isInItaly(X), not\ \ isRiver\ (X)$$

$$isPolitOfUSA(X) \leftarrow bornInUSA(X), isGov\ (X), not\ isPolitPuertoRico(X)$$

$$bornInUSA(X\ ) \leftarrow actedInMovie(X), createdMovie(X), not\ wonFilmfare(X)$$

Summary

- Conclusion
  - Quality-based theory revision under OWA
  - Partial materialization for ranking revisions
  - Comparison of ranking methods on real life KGs

- Outlook
  - Extending to higher arity predicates
    - Binary predicates [Tran et al., to appear ILP2016]
  - Evidence from text corpora
  - Exploiting partial completeness

References

- **[Angiulli and Fassetti, 2014]** Fabrizio Angiulli and Fabio Fassetti. Exploiting domain knowledge to detect outliers. *Data Min. Knowl. Discov.*, 28(2):519–568, 2014.
- **[Dimopoulos and Kakas, 1995]** Yannis Dimopoulos and Antonis C. Kakas. Learning non-monotonic logic programs: Learning exceptions. *In Machine Learning: ECML-95, 8th European Conference on Machine Learning*, Heraclion, Crete, Greece, April 25-27, 1995, Proceedings, pages 122–137, 1995.
- **[Galarraga et al., 2015]** Luis Galarraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. Fast Rule Mining in Ontological Knowledge Bases with AMIE+*. In VLDB Journal*, 2015.
- **[Law et al., 2015]** Mark Law, Alessandra Russo, and Krysia Broda. The ILASP system for learning answer set programs, 2015.
- **[Leone et al., 2006]** Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. 2006. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Logic 7*, 3 (July 2006), 499-562.
- **[Suzuki, 2006]** Einoshin Suzuki. Data mining methods for discovering interesting exceptions from an unsupervised table. *J. UCS*, 12(6):627–653, 2006.
- **[Tran et al., 2016]** Hai Dang Tran, Daria Stepanova, Mohamed H. Gad-Elrab, Francesca A. Lisi, Gerhard Weikum. Towards Nonmonotonic Relational Learning from Knowledge Graphs. *ILP2016*, London, UK, to appear.
- **[Katzouris et al., 2015]** Nikos Katzouris, Alexander Artikis, and Georgios Paliouras. Incremental learning of event definitions with inductive logic programming. Machine Learning, 100(2-3):555–585, 2015.

Related Work

- Learning nonmonotonic programs
  - E.g., [Dimopoulos and Kakas, 1995], ILASP [Law et al., 2015], ILED [Katzouris et al., 2015], etc.
- Outlier detection in logic programs
  - E.g., [Angiulli and Fassetti, 2014], etc.
- Mining exception rules
  - E.g., [Suzuki, 2006], etc.

Problem Statement: Ruleset Quality

- Independent **R**ule **M**easure (rm)
  - Support: $supp(H \leftarrow B) = supp(H \cup B)$
  - Coverage: $cov(H \leftarrow B) = supp(B)$
  - Confidence: $conf(H \leftarrow B) = \frac{supp(H \cup B)}{supp(B)}$
  - Lift: $lift(H \leftarrow B) = \frac{conf(H \leftarrow B)}{supp(H)}$
  - …
- Average Ruleset Quality

$$q_{rm}(R_{NM}, G) = \sum_{r \in R_{NM}} \frac{rm(r, G)}{|R_{NM}|}$$

Maximize

Problem Statement: Conflicting Predictions

- Measuring conflicts (auxiliary rules)

$$r_2: releasedInJP(X) \leftarrow hasJPComposer(X), not\ publisherUSA(X)$$

$$r_2^{aux}: not\_releasedInJP(X) \leftarrow hasJPComposer(X), publisherUSA(X)$$

$$q_{conflict}(R_{NM}, G) = \frac{|\{(p(a), not\_p(a)),\ \ldots\}|}{|\{not\_p(a),\ \ldots\}|}$$

Propositionalization

- Unary predicates

Binary Predicates

- hasType(enstein, scientist)
- isMarriedTo(elsa, einstein)
- bornIn(einstein, um)

Unary

- isAScientist(einstein)
- isMarriedToEinstein(elsa)
- bornInUlm(einstein)

Abstraction

- isAScientist(einstein)
- isMarriedToScientist(elsa)
- bornInGermany(einstein)

Experiments

- Input

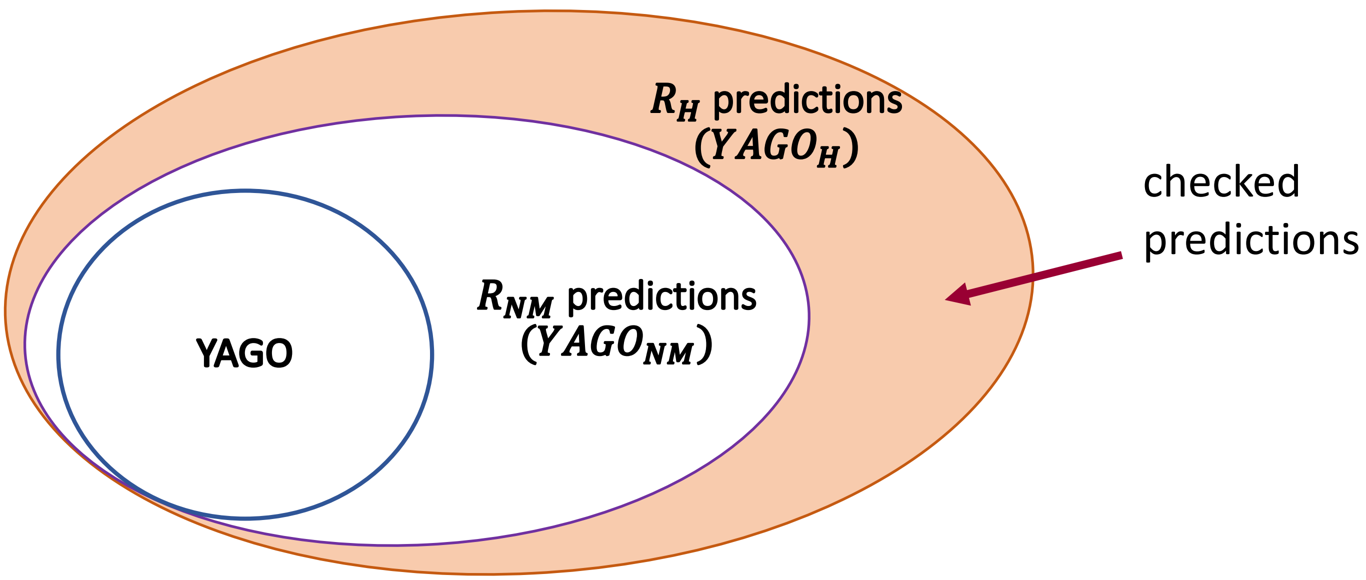
General-purpose KG **Domain-specific KG (Movies)**

- Experiment statistics

	YAGO3	IMDB
Input Facts	10M	2M
Horn Rules	10K	25K
Revised Rules	6K	22K

Experiments

- Predictions assessment
  - Run DLV on YAGO and R_H then R_{NM} seperately
  - Sample facts such that fact $f \in YAGO_H \backslash YAGO_{NM}$
  - 73% of the sampled facts were found to be erroneous



Ranking Rule's Revisions

- Partial materialization ranker
  - Augment the original KG with predictions of other rules
  - Rank revisions on Avg. confidence of the r and r^{aux}

$$score(r_e, KG^*) = \frac{conf(r_e, KG^*) + conf(r_e^{aux}, KG^*)}{2}$$

where r_e is the rule r with exception e & $KG^*$is the augmented KG.