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 }

? hasChild Barack Obama

Barack Obama hasChild Malia

Barack Obama hasChild Sasha
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.
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).
  - **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 \((s, r)\).
  - **PCA oracle**: \((s, r)\) 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

Complete instances in the domain of `hasChild`
Completeness oracles

- **CWA:** $cwa(s, r) = \text{true}$
- **PCA:** $pca(s, r) = \exists o : r(s, o)$
- **Cardinality:** $\text{card}(s, r) = \#(o : r(s, o)) \geq k$
- **Popular entities:** $\text{popularity}_{\text{pop}}(s, r) = \text{pop}(s)$
- **No-chg over time:** $\text{nochange}_{\text{chg}}(s, r) = \sim \text{chg}(s, r)$
- **Star:** $\text{star}_{r_1,\ldots,r_n}(s, r) = \forall i \in \{1,\ldots,n\} : \exists o : r_i(s, o)$
- **Class:** $\text{class}_c(s, r) = \text{type}(s, c)$
- **AMIE (rule mining)**
Completeness oracles

- **CWA**: \( \text{cwa}(s, r) = \text{true} \)
- **PCA**: \( \text{pca}(s, r) = \exists o : r(s, o) \)
- **Cardinality**: \( \text{card}(s, r) = \#(o : r(s, o)) \geq k \)
- **Popular entities**: \( \text{popularity}_{\text{pop}}(s, r) = \text{pop}(s) \)
- **No-chg over time**: \( \text{nochange}_{\text{chg}}(s, r) = \sim \text{chg}(s, r) \)
- **Star**: \( \text{star}_{r_1,\ldots,r_n}(s, r) = \forall i \in \{1,\ldots,n\} : \exists o : r_i(s, o) \)
- **Class**: \( \text{class}_c(s, r) = \text{type}(s, c) \)  
  Learned oracles
- **AMIE (rule mining)**
AMIE oracle

- It combines all the other oracles using rules.
AMIE oracle

• It combines all the other oracles using rules.

\[
\text{notype}(x, \text{Adult}), \text{type}(x, \text{Person}) \Rightarrow \text{complete}(x, \text{hasChild}) \quad \text{class}_{\text{non-adult}}(s, r)
\]

\[
\text{dateOfDeath}(x, y), \text{lessThan}_1(x, \text{placeOfDeath}) \Rightarrow \text{incomplete}(x, \text{placeOfDeath})
\]
AMIE oracle

- It combines all the other oracles using rules.

\[
\text{notype}(x, \text{Adult}), \text{type}(x, \text{Person}) \Rightarrow \text{complete}(x, \text{hasChild}) \quad \text{\text{class}}_{\text{non-adult}}(s, r)
\]

\[
\text{dateOfDeath}(x, y), \text{lessThan}_1(x, \text{placeOfDeath}) \Rightarrow \text{incomplete}(x, \text{placeOfDeath})
\]

- 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.

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

Predicting Completeness in Knowledge Bases
Luis Galárraga, Simon Razniekski, Antoine Amarilli, Fabian Suchanek

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 $q(K^*) \subseteq q(K)$.

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: $pop(s, r)$
  - $(s, r)$ is complete if $s$ is among the top 5% entities with most entities in the KB.
- No change: $nochange(s, r)$
  - $(s, r)$ is complete if the objects of $(s, r)$ have not changed w.r.t. a previous version of the KB.

**Parameterized**
- Star oracle: $star(s, r)$
  - $(s, r)$ is complete if we know object values for other properties $r_1, \ldots, r_t$ of $s$.
  - $producer(s, z), writer(x, w) \rightarrow complete(x, director)$
- Class oracle: $class(s, r)$
  - The KB is complete for entities in class $C$.
  - $Pope(x) \rightarrow complete(x, hasChild)$

**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.

- $President(x), moreThan(x, hasChild) \rightarrow complete(x, hasChild)$
- $dateOfDeath(x, y), lessThan(x, placeOfDeath) \rightarrow incomplete(x, placeOfDeath)$

Experimental evaluation

**FL-measure on YAGO**

<table>
<thead>
<tr>
<th>Property</th>
<th>AMIE</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>dateOfDeath</td>
<td>0%</td>
<td>2%</td>
<td>4%</td>
<td>6%</td>
<td>8%</td>
<td>10%</td>
<td>12%</td>
</tr>
<tr>
<td>hasChild</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>isPerson</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>isCitizen</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>isMarried</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>hasGender</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>hasParent</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>hasFrost</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>hasHappiness</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>hasIncome</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

---