On the Limits of Machine Knowledge: Completeness, Recall and Negation in Web-scale Knowledge Bases

Simon Razniewski, Hiba Arnaout, Shrestha Ghosh, Fabian Suchanek
On the Limits of Machine Knowledge: Completeness, Recall and Negation in Web-scale Knowledge Bases

Simon Razniewski, Hiba Arnaout, Shrestha Ghosh, Fabian Suchanek

1. Introduction & Foundations (Simon) – 20 min
2. Predictive recall assessment (Fabian) – 20 min
3. Counts from text and KB (Shrestha) – 20 min
4. Negation (Hiba) – 20 min
5. Wrap-up (Simon) – 5 min

https://www.mpi-inf.mpg.de/vldb-2021-tutorial
Machine knowledge in action
Machine knowledge in action

Knowledge-powered
Machine knowledge in action
Machine knowledge is awesome

• **Reusable, scrutable asset for knowledge-centric tasks**
  • Semantic search & QA
  • Entity-centric text analytics
  • Distant supervision for ML
  • Data cleaning

• **Impactful projects** at major commercial and public players
  • Wikidata, Google KG, Microsoft Satori, ...

• **Strongly rooted in database community**
  • Data integration, data cleaning, conceptual modelling, storage, indexing and querying, ...
But: Machine Knowledge is incomplete
Machine knowledge is incomplete (2)

Wikidata KB:

VLDB journal has only published 80 articles ever
  • https://scholia.toolforge.org/venue/Q15760089

Most cited papers on data integration have <38 citations
  • https://scholia.toolforge.org/topic/Q386824
But: Machine knowledge is one-sided

• In KB:
  • *Stephen Hawking won Presidential medal of freedom*
  • *Vietnam is a member of ASEAN*
  • *iPhone has 12MP camera*

• Not in KB:
  • *Stephen Hawking did not win the Nobel Prize*
  • *Switzerland is not a member of the EU*
  • *iPhone has no headphone jack*
Why is this problematic? (1) Querying

- Decision making more and more data-driven

- Analytical queries paint wrong picture of reality
  - *E.g., VLDB journal deemed too small*

- Instance queries return wrong results
  - *E.g., wrongly assuming certain authors never published in VLDBJ*
Why is this problematic? (1)

Data Curation

• Effort prioritization fundamental challenge in human-in-the-loop curation
  • *Should we spend effort on obtaining data for VLDB or TKDE?*

• Risk of effort duplication if not keeping track of completed areas
  • *Data for TKDE complete up to 2020*
Why is this problematic? (3)

Summarization and decision making

No free WiFi!

No headphone jack
How to know how much a KB knows?

How to = techniques
How much knows = completeness/recall/coverage estimation
KB = General world knowledge repository
What this tutorial offers

• **Logical foundations**
  • Languages for describing KB completeness (part 1)

• **Predictive assessment**
  • How (in-)completeness can be statistically predicted (Part 2)

• **Count information**
  • How count information enables (in-)completeness assessment (Part 3)

• **Negation**
  • How salient negations can be derived from incomplete KBs (Part 4)

**Goals:**
1. Systematize the topic and its facets
2. Lay out assumptions, strengths and limitations of approaches
3. Provide a practical toolsuite
Relevant research domains

• Databases
• Logics
• Statistics
• Machine Learning
• Natural language processing
What this tutorial is NOT about

• Knowledge base completion (KBC)
  • “How to make KBs more complete”

• Related: Understanding of completeness is needed to know when/when not to employ KBC
  • KBC naively is open-ended
    → Understanding of completeness needed to “stop”

• But:
  • Heuristic, error-prone KBC not always desired
  • Completeness awareness != actionable completion

• Literature on knowledge graph completion, link prediction, missing value imputation, etc.
  • E.g., Rossi, Andrea, et al. Knowledge graph embedding for link prediction: A comparative analysis. TKDD 2021
On the Limits of Machine Knowledge: Completeness, Recall and Negation in Web-scale Knowledge Bases

Simon Razniewski, Hiba Arnaout, Shrestha Ghosh, Fabian Suchanek

1. **Introduction & Foundations** (Simon) – 20 min
2. Predictive recall assessment (Fabian) – 20 min
3. Counts from text and KB (Shrestha) – 20 min
4. Negation (Hiba) – 20 min
5. Wrap-up (Simon) – 5 min
Knowledge base - definition

Given set \( E \) (entities), \( L \) (literals), \( P \) (predicates)

- Predicates are positive or negated properties
  - \( bornIn, notWonAward, \ldots \)

- An assertion is a triple \( (s, p, o) \in E \times P \times (E \cup L) \)

- An available KB \( K^a \) is a set of assertions

- The \"ideal\" (complete) KB is called \( K^i \)

- Available KBs are incomplete: \( K^a \subseteq K^i \)
Knowledge bases (KBs aka. KGs)

subject-predicate-object triples about entities, attributes of and relations between entities

predicates (subject, object)

type (Marie Curie, physicist)
subtypeOf (physicist, scientist)

placeOfBirth (Marie Curie, Warsaw)
residence (Marie Curie, Paris)
¬placeOfBirth (Marie Curie, France)

discovery (Polonium, 12345)
discoveryDate (12345, 1898)
discoveryPlace (12345, Paris)
discoveryPerson (12345, Marie Curie)

atomicNumber (Polonium, 84)
halfLife (Polonium, 2.9 y)

+ composite objects

taxonomic knowledge

factual knowledge

spatio-temporal & contextual knowledge

expert knowledge
History of knowledge bases

**Cyc**

**WordNet**

Manual compilation

Automation and human-in-the-loop

Wikipedia

6 Mio. English articles
40 Mio. contributors

1985
1990
2000
2005
2010
2020

\( \forall x: \text{human}(x) \Rightarrow (\exists y: \text{mother}(x,y) \land \exists z: \text{father}(x,z)) \)

\( \forall x,u,w: (\text{mother}(x,u) \land \text{mother}(x,w) \Rightarrow u=w) \)

\( \text{guitarist} \subseteq \{\text{player, musician}\} \subseteq \text{artist} \subseteq \{\text{player, footballer}\} \subseteq \text{athlete} \)
KB scale and use cases

**Wikidata** (open)
- 95 M items
- 1.1 B statements

**Google KG**
- 5 B items
- 500 B statements

**Major use cases:**
- semantic search & QA
- language understanding
- distant supervision for ML
- data cleaning
KB incompleteness is inherent

Einstein received the Nobel Prize in 1921, the Copley medal, the Prix Jules Jansen, the Medal named after Max Planck, and several others.

1. Sources incomplete

Knowledge base construction

2. Extractors imperfect

Award(Einstein, Nobel Prize)
Award(Einstein, Copley medal)
Award(Einstein, Prix Jules Jansen)
Friend(Einstein, Max Planck)

3. Extraction resource-bounded

Honorary doctorate, UMadrid
Gold medal, Royal Astronomic Society
Benjamin Franklin Medal,
...

Weikum et al.
Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases
FnT 2021
Root challenges

1. Available KBs are incomplete
   \[ K^a < K^i \]

2. Available KBs hardly store negatives
   \[ K^a^- \approx \emptyset \]
Formal semantics for incomplete KBs: Closed and open-world assumption

<table>
<thead>
<tr>
<th>won</th>
<th>name</th>
<th>award</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>award</td>
<td></td>
</tr>
<tr>
<td>Brad Pitt</td>
<td>Oscar</td>
<td></td>
</tr>
<tr>
<td>Einstein</td>
<td>Nobel Prize</td>
<td></td>
</tr>
<tr>
<td>Berners-Lee</td>
<td>Turing Award</td>
<td></td>
</tr>
</tbody>
</table>

Closed-world assumption

- \( \text{won(} \text{Brad Pitt}, \text{Oscar)} \)? \( \rightarrow \text{Yes} \)
- \( \text{won(} \text{Pitt}, \text{Nobel Prize)} \)? \( \rightarrow \text{No} \)

Open-world assumption

- \( \text{won(} \text{Brad Pitt}, \text{Oscar)} \)? \( \rightarrow \text{Yes} \)
- \( \text{won(} \text{Pitt}, \text{Nobel Prize)} \)? \( \rightarrow \text{Maybe} \)

- Databases traditionally employ closed-world assumption
- KBs (semantic web) necessarily operate under open-world assumption
Open-world assumption

World-aware AI?
Practically useful paradigm?

- Q: Game of Thrones directed by Shakespeare?
  KB: Maybe

- Q: Brad Pitt attends VLDB?
  KB: Maybe

- Q: Trump brother of Kim Jong Un?
  KB: Maybe
The logicians way out – completeness assertions

• Need power to express both *maybe* and *no*  
  (*Some paradigm which allows both open- and closed-world interpretation of data to co-exist*)

• Approach: *Completeness statements* [Motro 1989]

<table>
<thead>
<tr>
<th>won</th>
<th>Completeness statement:</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>award</td>
</tr>
<tr>
<td>Brad Pitt</td>
<td>Oscar</td>
</tr>
<tr>
<td>Einstein</td>
<td>Nobel Prize</td>
</tr>
<tr>
<td>Berners-Lee</td>
<td>Turing Award</td>
</tr>
</tbody>
</table>

- **wonAward** is complete for **Nobel Prizes**
  - **won(Pitt, Oscar)?** → *Yes*
  - **won(Pitt, Nobel)?** → *No* (CWA)
  - **won(Pitt, Turing)?** → *Maybe* (OWA)
The power of completeness assertions

Know what the KB knows:

→ Locally, $K^a = K^i$

Absent assertions are really false:

→ Locally, $s \not\in K^a$ implies $s \not\in K^i$
Completeness statements: Formal view

Complete ( won(name, award); award = ‘Nobel’)

Implies constraint on possible state of $K^a$ and $K^i$

\[ \text{won}^i(name, ‘Nobel’) \rightarrow \text{won}^a(name, ‘Nobel’) \]
(tuple-generating dependency)
Cardinality assertions: Formal view

• “Nobel prize was awarded 603 times”
  → \(|\text{won}^i(\text{name, ‘Nobel’})| = 603\)

→ Allows counting objects in \(K^a\)
  • Equivalent count → Completeness assertion
  • Otherwise, fractional coverage/recall information
    • “93% of awards covered”

• Grounded in number restrictions/role restrictions in Description Logics

B. Hollunder and F. Baader
*Qualifying Number Restrictions in Concept Languages*
KR 1991
Formal reasoning with completeness assertions

**Problem:** Query completeness reasoning

**Input:**
- Set of completeness assertions for base relations
- Query Q

**Task:**
- Compute completeness assertions that hold for result of Q
Formal reasoning with completeness assertions

<table>
<thead>
<tr>
<th>Work</th>
<th>Description Language</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motro, TODS 1989</td>
<td>Views</td>
<td>Algorithm</td>
</tr>
<tr>
<td>Fan &amp; Geerts, PODS 2009</td>
<td>Various query languages (CQ-Datalog)</td>
<td>Decidability/Complexity</td>
</tr>
<tr>
<td>Razniewski &amp; Nutt 2011</td>
<td>Join queries</td>
<td>Complexity</td>
</tr>
<tr>
<td>Lang et al., SIGMOD 2014</td>
<td>Selections</td>
<td>Algorithm</td>
</tr>
<tr>
<td>Razniewski et al., SIGMOD 2016</td>
<td>Selections</td>
<td>Algorithm, computational completeness</td>
</tr>
</tbody>
</table>
Where can completeness statements come from?

- Data creators should pass them along as metadata
- Or editors should add them in curation steps

<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abingdon</td>
<td>Residential triangle, Longmead etc.</td>
<td>Pub is only restaurant? Footways that link stuff, stubbed in places.</td>
</tr>
<tr>
<td>Shippon</td>
<td>Whole village, minus the barracks</td>
<td>Mostly done here.</td>
</tr>
</tbody>
</table>

- E.g., COOL-WD tool (Completeness tool for Wikidata)
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>residence (P551)</td>
<td>White House</td>
</tr>
<tr>
<td>country of citizenship (P27)</td>
<td>United States of America</td>
</tr>
<tr>
<td>child (P40)</td>
<td>Ivanka Trump</td>
</tr>
<tr>
<td></td>
<td>Donald Trump Jr.</td>
</tr>
<tr>
<td></td>
<td>Eric Trump</td>
</tr>
<tr>
<td></td>
<td>Tiffany Trump</td>
</tr>
<tr>
<td></td>
<td>Barron Trump</td>
</tr>
<tr>
<td>field of work (P101)</td>
<td>politics</td>
</tr>
<tr>
<td></td>
<td>government</td>
</tr>
</tbody>
</table>
But...

- Requires human effort
  - Editors are lazy
  - Automatically created KBs do not even have editors

Remainder of this tutorial:

How to automatically acquire information about what a KB knows
Takeaway Part 1: Foundations

• KBs are pragmatic collections of knowledge
  • Issue 1: Inherently incomplete
  • Issue 2: Hardly store negative knowledge

• Open-world assumption (OWA) as formal interpretation leads to counterintuitive results

• Metadata about completeness or counts as way out

Next: How to use predictive models for completeness assessment
On the Limits of Machine Knowledge: Completeness, Recall and Negation in Web-scale Knowledge Bases

Simon Razniewski, Hiba Arnaout, Shrestha Ghosh, Fabian Suchanek

1. Introduction & Foundations (Simon) – 20 min
2. Predictive recall assessment (Fabian) – 20 min
3. Counts from text and KB (Shrestha) – 20 min
4. Negation (Hiba) – 20 min
5. Wrap-up (Simon) – 5 min
Wrap-up: Take-aways

1. KBs are *incomplete* and *limited* on the *negative* side

2. Predictive techniques work from a surprising set of paradigms

3. Count information a prime way to gain insights into completeness/coverage

4. Salient negations can be heuristically *materialized*
Wrap-up: Recipe

• Ab-initio KB construction
  • Intertwine data and metadata collection
  • Human insertion: Provide tools
  • Automated extraction: Learn from extraction context

• KB curation
  • Exploit KB-internal or textual cardinality assertions
  • Inspect statistical properties on density or distribution
  • Compute overlaps on pseudo-random samples
Open research questions

1. How are entity, property and fact completeness related?

2. How to distinguish salient negations from data modelling issues?

3. How to estimate coverage of knowledge in pre-trained language models?
Wrap-up: Wrap-up

• KBs major drivers of knowledge-intensive applications

• Severe limitations concerning completeness and coverage-awareness

• This tutorial: Overview of problem, techniques and tools to obtain awareness of completeness
Takeaway Part 1: Foundations

- KBs are pragmatic collections of knowledge
  - Issue 1: Inherently incomplete
  - Issue 2: Hardly store negative knowledge

- Open-world assumption (OWA) as formal interpretation leads to counterintuitive results

- Metadata about completeness or counts as way out

Takeaway: Predictive recall assessment

Using statistical techniques, we can predict more or less

- the recall of facts
  - are we missing objects for a subject?
  - do all subjects have an attribute in the real world?
  - does a text enumerate all objects for a subject?

- the recall of entities
  - is the distribution of entities representative?
  - how many entities are in the real world?

Takeaway: Counts from text and KB

1. Count information comes in two variants
   - Counting predicates - store integer counts
   - Enumerating predicates - store entities
2. Count information in text
   - occurs as cardinals, ordinals, non-numeric noun phrases
   - occurs with compositional cues
3. Count information in KBs
   - is expressed in two variants
   - occurs semantically related count predicates
4. Count information
   - can enrich KB
   - highlight inconsistencies

Takeaway: negation

- Current KBs lack negative knowledge

- Rising interest in the explicit addition of negation to OW KB.

- Negations highly relevant in many applications including:
  - Commercial decision making (e.g., hotel booking)
  - General-domain question answering systems (e.g., is Switzerland a member of the EU?)

- Methodologies include:
  - Statistical inference
  - Text extraction
  - Pretrained LMs.