Knowledge Expansion over Probabilistic Knowledge Bases

Yang Chen, Daisy Zhe Wang
{yang, daisyw}@cise.ufl.edu

Computer and Information Science and Engineering
University of Florida

SIGMOD, Snowbird, UT
Jun 25, 2014
Outline

1. Introduction
   - Knowledge Bases
   - Knowledge Expansion

2. The ProbKB System
   - Probabilistic Knowledge Bases
   - ProbKB Architecture
   - Grounding
   - Quality Control

3. Conclusion
   - Conclusion
Knowledge Bases

- A knowledge base is a collection of entities, facts, and relationships that conforms with a certain data model.
- Allows machines to interpret human information in a principled manner.
Knowledge Bases

- A *knowledge base* is a collection of entities, facts, and relationships that conforms with a certain data model.
- Allows machines to interpret human information in a principled manner.

**Figure**: Google knowledge graph
Knowledge Bases

- A *knowledge base* is a collection of entities, facts, and relationships that conforms with a certain data model.
- Allows machines to interpret human information in a principled manner.
- But they are often *incomplete*.

*Figure: Google knowledge graph*
Knowledge Base Construction Review

1. Human collaboration:
   - DBPedia, Freebase, Google Knowledge Graph, YAGO.
2. Automatic construction:
   - DeepDive, Knowledge Vault, NELL, OpenIE, ProBase, YAGO.
3. Knowledge integration:
   - Knowledge Fusion, Knowledge Vault, PIDGIN.
Inferring Implicit Information

**Nutritional value** [edit]

Kale is very high in *beta carotene*, *vitamin K*, *vitamin C*, and rich in *calcium*. Kale is a source of two carotenoids, *lutein* and *zeaxanthin*. Kale, as with *broccoli* and other *brassicas*, contains *sulforaphane* (particularly when chopped or minced), a chemical with potent anti-cancer properties.

Boiling decreases the level of *sulforaphane*; however, *steaming*, *microwaving*, or *stir frying* does not result in significant loss. Along with other brassica vegetables, kale is also a source of *indole-3-carbinol*, a chemical which boosts DNA repair in cells and appears to block the growth of cancer cells. Kale has been found to contain a group of resins known as *bile acid sequestrants*, which have been shown to lower cholesterol and decrease absorption of dietary fat. Steaming significantly increases these bile acid binding properties.

The risk of osteoporosis fractures can be reduced with lifestyle changes and in those with previous osteoporosis related fractures, medications. Lifestyle change includes diet, exercise, and preventing falls. The utility of *calcium* and *vitamin D* is questionable in most. Bisphosphonates are useful in those with previous fractures from osteoporosis but are of minimal benefit in those who have osteoporosis but no previous fractures. Osteoporosis is a component of the *frailty syndrome*.

- Kale is rich in Calcium ∧ Calcium helps prevent Osteoporosis → Kale helps prevent Osteoporosis.
Inferring Implicit Information

Nutritional value [edit]

Kale is very high in beta carotene, vitamin K, vitamin C, and rich in calcium. Kale is a source of two carotenoids, lutein and zeaxanthin. Kale, as with broccoli and other brassicas, contains sulforaphane (particularly when chopped or minced), a chemical with potent anti-cancer properties. Boiling decreases the level of sulforaphane; however, steaming, microwaving, or stir frying does not result in significant loss. Along with other brassica vegetables, kale is also a source of indole-3-carbinol, a chemical which boosts DNA repair in cells and appears to block the growth of cancer cells. Kale has been found to contain a group of resins known as bile acid sequestrants, which have been shown to lower cholesterol and decrease absorption of dietary fat. Steaming significantly increases these bile acid binding properties.

The risk of osteoporosis fractures can be reduced with lifestyle changes and in those with previous osteoporosis related fractures, medications. Lifestyle change includes diet, exercise, and preventing falls. The utility of calcium and vitamin D is questionable in most. Bisphosphonates are useful in those with previous fractures from osteoporosis but are of minimal benefit in those who have osteoporosis but no previous fractures. Osteoporosis is a component of the frailty syndrome.

- Kale is rich in Calcium \( \land \) Calcium helps prevent Osteoporosis → Kale helps prevent Osteoporosis.
Inferring Implicit Information

\[
\text{IsHeadquarteredIn}(\text{Company}, \text{State}) :- \\
\quad \text{IsBasedIn}(\text{Company}, \text{City}) \land \text{IsLocatedIn}(\text{City}, \text{State}); \\
\text{Contains}(\text{Food}, \text{Chemical}) :- \\
\quad \text{IsMadeFrom}(\text{Food}, \text{Ingredient}) \land \text{Contains}(\text{Ingredient}, \text{Chemical}); \\
\text{Reduce}(\text{Medication}, \text{Factor}) :- \\
\quad \text{KnownGenericallyAs}(\text{Medication}, \text{Drug}) \land \text{Reduce}(\text{Drug}, \text{Factor}); \\
\text{ReturnTo}(\text{Writer}, \text{Place}) :- \\
\quad \text{BornIn}(\text{Writer}, \text{City}) \land \text{CapitalOf}(\text{City}, \text{Place}); \\
\text{Make}(\text{Company1}, \text{Device}) :- \\
\quad \text{Buy}(\text{Company1}, \text{Company2}) \land \text{Make}(\text{Company2}, \text{Device});
\]

**Figure:** SHERLOCK Horn clauses learner.
Contributions

Knowledge Expansion Problem

Inferring implicit knowledge in KBs.

- Efficiency.
  - We use DBMSes to model knowledge bases;
  - We design a SQL-based algorithm to apply inference rules *in batches*;
  - We use MPP databases to parallelize the inference process.

- Quality.
  - We identify major error sources and combine state-of-the-art methods to detect and recover from errors;
  - We use semantic constraints to identify errors and ambiguities;
  - We clean the rule set based on their statistical properties.
Contributions

Knowledge Expansion Problem
Inferring implicit knowledge in KBs.

- **Efficiency.**
  - We use DBMSes to model knowledge bases;
  - We design a SQL-based algorithm to apply inference rules *in batches*;
  - We use MPP databases to parallelize the inference process.

- **Quality.**
  - We identify major error sources and combine state-of-the-art methods to detect and recover from errors;
  - We use semantic constraints to identify errors and ambiguities;
  - We clean the rule set based on their statistical properties.
Contributions

Knowledge Expansion Problem
Inferring implicit knowledge in KBs.

- Efficiency.
  - We use DBMSes to model knowledge bases;
  - We design a SQL-based algorithm to apply inference rules in batches;
  - We use MPP databases to parallelize the inference process.

- Quality.
  - We identify major error sources and combine state-of-the-art methods to detect and recover from errors;
  - We use semantic constraints to identify errors and ambiguities;
  - We clean the rule set based on their statistical properties.
Contributions

Knowledge Expansion Problem
Inferring implicit knowledge in KBs.

- **Efficiency.**
  - We use DBMSes to model knowledge bases;
  - We design a SQL-based algorithm to apply inference rules *in batches*;
  - We use MPP databases to parallelize the inference process.

- **Quality.**
  - We identify major error sources and combine state-of-the-art methods to detect and recover from errors;
  - We use semantic constraints to identify errors and ambiguities;
  - We clean the rule set based on their statistical properties.
Contributions

Knowledge Expansion Problem

Inferring implicit knowledge in KBs.

- Efficiency.
  - We use DBMSes to model knowledge bases;
  - We design a SQL-based algorithm to apply inference rules *in batches*;
  - We use MPP databases to parallelize the inference process.

- Quality.
  - We identify major error sources and combine state-of-the-art methods to detect and recover from errors;
  - We use semantic constraints to identify errors and ambiguities;
  - We clean the rule set based on their statistical properties.
Contributions

Knowledge Expansion Problem
Inferring implicit knowledge in KBs.

- Efficiency.
  - We use DBMSes to model knowledge bases;
  - We design a SQL-based algorithm to apply inference rules *in batches*;
  - We use MPP databases to parallelize the inference process.

- Quality.
  - We identify major error sources and combine state-of-the-art methods to detect and recover from errors;
  - We use semantic constraints to identify errors and ambiguities;
  - We clean the rule set based on their statistical properties.
Contributions

Knowledge Expansion Problem
Inferring implicit knowledge in KBs.

Efficiency.
- We use DBMSes to model knowledge bases;
- We design a SQL-based algorithm to apply inference rules in batches;
- We use MPP databases to parallelize the inference process.

Quality.
- We identify major error sources and combine state-of-the-art methods to detect and recover from errors;
- We use semantic constraints to identify errors and ambiguities;
- We clean the rule set based on their statistical properties.
Contributions

Knowledge Expansion Problem
Inferring implicit knowledge in KBs.

- **Efficiency.**
  - We use DBMSes to model knowledge bases;
  - We design a SQL-based algorithm to apply inference rules *in batches*;
  - We use MPP databases to parallelize the inference process.

- **Quality.**
  - We identify major error sources and combine state-of-the-art methods to detect and recover from errors;
  - We use semantic constraints to identify errors and ambiguities;
  - We clean the rule set based on their statistical properties.
Contributions

Knowledge Expansion Problem
Inferring implicit knowledge in KBs.

- Efficiency.
  - We use DBMSes to model knowledge bases;
  - We design a SQL-based algorithm to apply inference rules in batches;
  - We use MPP databases to parallelize the inference process.

- Quality.
  - We identify major error sources and combine state-of-the-art methods to detect and recover from errors;
  - We use semantic constraints to identify errors and ambiguities;
  - We clean the rule set based on their statistical properties.
Outline

1. Introduction
   - Knowledge Bases
   - Knowledge Expansion

2. The ProbKB System
   - Probabilistic Knowledge Bases
   - ProbKB Architecture
   - Grounding
   - Quality Control

3. Conclusion
   - Conclusion
Probabilistic Knowledge Bases

Example (Probabilistic Knowledge Bases)

We define a *probabilistic knowledge base* to be a 5-tuple $\Gamma = (\mathcal{E}, \mathcal{C}, \mathcal{R}, \Pi, \mathcal{L})$:

<table>
<thead>
<tr>
<th>Entities $\mathcal{E}$</th>
<th>Classes $\mathcal{C}$</th>
<th>Relations $\mathcal{R}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ruth Gruber, New York City, Brooklyn</td>
<td>$W$ (Writer) = {Ruth Gruber}, $C$ (City) = {New York City}, $P$ (Place) = {Brooklyn}</td>
<td>born in($W, P$), born in($W, C$), live in($W, P$), live in($W, C$), locate in($P, C$)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Facts $\Pi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.93 born in(Ruth Gruber, Brooklyn)</td>
</tr>
<tr>
<td>0.96 born in(Ruth Gruber, New York City)</td>
</tr>
</tbody>
</table>
## Probabilistic Knowledge Bases

### Example (ReVerb-Sherlock KB Cont.)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.40</td>
<td>$\forall x \in W \forall y \in P$ $(\text{live in}(x, y) \leftarrow \text{born in}(x, y))$</td>
</tr>
<tr>
<td>1.53</td>
<td>$\forall x \in W \forall y \in C$ $(\text{live in}(x, y) \leftarrow \text{born in}(x, y))$</td>
</tr>
<tr>
<td>2.68</td>
<td>$\forall x \in W \forall y \in P$ $(\text{grow up in}(x, y) \leftarrow \text{born in}(x, y))$</td>
</tr>
<tr>
<td>0.74</td>
<td>$\forall x \in W \forall y \in C$ $(\text{grow up in}(x, y) \leftarrow \text{born in}(x, y))$</td>
</tr>
<tr>
<td>0.32</td>
<td>$\forall x \in P \forall y \in C \forall z \in W$ $(\text{locate in}(x, y) \leftarrow \text{live in}(z, x) \land \text{live in}(z, y))$</td>
</tr>
<tr>
<td>0.52</td>
<td>$\forall x \in P \forall y \in C \forall z \in W$ $(\text{locate in}(x, y) \leftarrow \text{born in}(z, x) \land \text{born in}(z, y))$</td>
</tr>
<tr>
<td>$\infty$</td>
<td>$\forall x \in C \forall y \in C \forall z \in W$ $(\text{born in}(z, x) \land \text{born in}(z, y) \rightarrow x = y)$</td>
</tr>
</tbody>
</table>

**Table:** Probabilistic KB from ReVerb-Sherlock.
MLN: The State-of-the-Art

DB ↔ Program ↔ MLN
(Weighted rules)
**ProbKB In-Database Architecture**

- **Inference Engine (e.g., GraphLab)**
- **RDMBS**
  - Factor Graph
  - SQL
  - UDF/UDA
- **Query Optimizer & Execution Engine**
  - MLN
  - Entities
  - Facts

---

Knowledge Expansion over Probabilistic Knowledge Bases
**ProbKB In-Database Architecture**

- **RDMBS**
  - Factor Graph
  - SQL
  - UDF/UDA
  - Query Optimizer & Execution Engine
  - MLN
  - Entities
  - Facts
Relational ProbKB

| born in(Ruth Gruber, Brooklyn) | 0.93 |
| born in(Ruth Gruber, New York City) | 0.96 |
## Relational ProbKB

<table>
<thead>
<tr>
<th>$I$</th>
<th>$R$</th>
<th>$x$</th>
<th>$C_1$</th>
<th>$y$</th>
<th>$C_2$</th>
<th>$w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>born in</td>
<td>RG</td>
<td>$W$</td>
<td>Br</td>
<td>$P$</td>
<td>0.93</td>
</tr>
<tr>
<td>2</td>
<td>born in</td>
<td>RG</td>
<td>$W$</td>
<td>NYC</td>
<td>$C$</td>
<td>0.96</td>
</tr>
</tbody>
</table>
Relational ProbKB

Definition

Two first-order clauses are defined to be *structurally equivalent* if they differ only in the entities, classes, and relations symbols.

\[
\forall x \in W \forall y \in P \ (\text{live in}(x, y) \leftarrow \text{born in}(x, y)) \quad 1.40 \\
\forall x \in W \forall y \in C \ (\text{live in}(x, y) \leftarrow \text{born in}(x, y)) \quad 1.53 \\
\forall x \in W \forall y \in P \ (\text{grow up in}(x, y) \leftarrow \text{born in}(x, y)) \quad 2.68 \\
\forall x \in W \forall y \in C \ (\text{grow up in}(x, y) \leftarrow \text{born in}(x, y)) \quad 0.74
\]

<table>
<thead>
<tr>
<th>(R_1)</th>
<th>(R_2)</th>
<th>(C_1)</th>
<th>(C_2)</th>
<th>(w)</th>
</tr>
</thead>
<tbody>
<tr>
<td>live in</td>
<td>born in</td>
<td>(W)</td>
<td>(P)</td>
<td>1.40</td>
</tr>
<tr>
<td>live in</td>
<td>born in</td>
<td>(W)</td>
<td>(C)</td>
<td>1.53</td>
</tr>
<tr>
<td>grow up in</td>
<td>born in</td>
<td>(W)</td>
<td>(P)</td>
<td>2.68</td>
</tr>
<tr>
<td>grow up in</td>
<td>born in</td>
<td>(W)</td>
<td>(C)</td>
<td>0.74</td>
</tr>
</tbody>
</table>
Relational ProbKB

Definition

Two first-order clauses are defined to be *structurally equivalent* if they differ only in the entities, classes, and relations symbols.

\[
\forall x \in W \ \forall y \in P \ (\text{live in}(x, y) \leftarrow \text{born in}(x, y)) \quad 1.40
\]

\[
\forall x \in W \ \forall y \in C \ (\text{live in}(x, y) \leftarrow \text{born in}(x, y)) \quad 1.53
\]

\[
\forall x \in W \ \forall y \in P \ (\text{grow up in}(x, y) \leftarrow \text{born in}(x, y)) \quad 2.68
\]

\[
\forall x \in W \ \forall y \in C \ (\text{grow up in}(x, y) \leftarrow \text{born in}(x, y)) \quad 0.74
\]
Relational \textbf{ProbKB}

\textbf{Definition}

Two first-order clauses are defined to be \textit{structurally equivalent} if they differ only in the entities, classes, and relations symbols.

\[
\forall x \in P \ \forall y \in C \ \forall z \in W \ (\text{locate in}(x, y) \leftarrow \text{live in}(z, x) \land \text{live in}(z, y)) \quad 0.32 \\
\forall x \in P \ \forall y \in C \ \forall z \in W \ (\text{locate in}(x, y) \leftarrow \text{born in}(z, x) \land \text{born in}(z, y)) \quad 0.52
\]
Relational ProbKB

Definition

Two first-order clauses are defined to be *structurally equivalent* if they differ only in the entities, classes, and relations symbols.

∀x ∈ P ∀y ∈ C ∀z ∈ W (locate in(x, y) ← live in(z, x) ∧ live in(z, y)) 0.32
∀x ∈ P ∀y ∈ C ∀z ∈ W (locate in(x, y) ← born in(z, x) ∧ born in(z, y)) 0.52

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
R_1 & R_2 & R_3 & C_1 & C_2 & C_3 & w \\
\hline
\text{located in} & \text{live in} & \text{live in} & P & C & W & 0.32 \\
\text{located in} & \text{born in} & \text{born in} & P & C & W & 0.52 \\
\hline
\end{array}
\]

M_3
Grounding

```sql
SELECT M1.R1 AS R,
    T.x AS x, T.C1 AS C1,
    T.y AS y, T.C2 AS C2
FROM M1
JOIN T ON M1.R2 = T.R
```

<table>
<thead>
<tr>
<th>( I )</th>
<th>( R )</th>
<th>( x )</th>
<th>( C_1 )</th>
<th>( y )</th>
<th>( C_2 )</th>
<th>( w )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>born in</td>
<td>RG</td>
<td>( W )</td>
<td>Br</td>
<td>( P )</td>
<td>0.93</td>
</tr>
<tr>
<td>2</td>
<td>born in</td>
<td>RG</td>
<td>( W )</td>
<td>NYC</td>
<td>( C )</td>
<td>0.96</td>
</tr>
<tr>
<td>3</td>
<td>live in</td>
<td>RG</td>
<td>( W )</td>
<td>NYC</td>
<td>( C )</td>
<td>0.96</td>
</tr>
<tr>
<td>4</td>
<td>grow up in</td>
<td>RG</td>
<td>( W )</td>
<td>Br</td>
<td>( P )</td>
<td>0.93</td>
</tr>
<tr>
<td>5</td>
<td>live in</td>
<td>RG</td>
<td>( W )</td>
<td>Br</td>
<td>( P )</td>
<td>0.74</td>
</tr>
<tr>
<td>6</td>
<td>grow up in</td>
<td>RG</td>
<td>( W )</td>
<td>Br</td>
<td>( P )</td>
<td>0.93</td>
</tr>
</tbody>
</table>

\( M_1 \)
Grounding

\[\text{SELECT } M1.R1 \text{ AS } R,\]
\[T.x \text{ AS } x, T.C1 \text{ AS } C1,\]
\[T.y \text{ AS } y, T.C2 \text{ AS } C2\]
\[\text{FROM } M1\]
\[\text{JOIN } T \text{ ON } M1.R2 = T.R\]
\[\text{AND } M1.C1 = T.C1 \text{ AND } M1.C2 = T.C2;\]
Grounding

\[
\begin{align*}
\text{SELECT} & \quad M1.R1 \quad \text{AS} \quad R, \\
& \quad T.x \quad \text{AS} \quad x, \quad T.C1 \quad \text{AS} \quad C1, \\
& \quad T.y \quad \text{AS} \quad y, \quad T.C2 \quad \text{AS} \quad C2 \\
\text{FROM} & \quad M1 \\
& \quad \text{JOIN} \quad T \quad \text{ON} \quad M1.R2 = T.R \\
& \quad \quad \quad \text{AND} \quad M1.C1 = T.C1 \quad \text{AND} \quad M1.C2 = T.C2;
\end{align*}
\]

\begin{table}
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
\textbf{} & \textbf{\(R_1\)} & \textbf{\(R_2\)} & \textbf{\(C_1\)} & \textbf{\(C_2\)} & \textbf{\(w\)} \\
\hline
1 & born in & RG & \(W\) & Br & \(P\) & 0.93 \\
2 & born in & RG & \(W\) & NYC & \(C\) & 0.96 \\
3 & live in & RG & \(W\) & NYC & \(C\) & 0.96 \\
4 & grow up in & RG & \(W\) & Br & \(C\) & 0.74 \\
5 & live in & RG & \(W\) & NYC & \(C\) & 0.74 \\
6 & grow up in & RG & \(W\) & Br & \(P\) & 0.74 \\
\hline
\end{tabular}
\end{table}
Grounding

\[
\text{SELECT } M1.R1 \text{ AS } R, \\
T.x \text{ AS } x, T.C1 \text{ AS } C1, \\
T.y \text{ AS } y, T.C2 \text{ AS } C2 \\
\text{FROM } M1 \\
\text{JOIN } T \text{ ON } M1.R2 = T.R \\
\text{AND } M1.C1 = T.C1 \text{ AND } M1.C2 = T.C2;
\]

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}
\hline
I & R & \Delta & C_1 & \Delta & C_2 & \Delta & w \\
\hline
1 & \text{born in} & RG & W & \text{Br} & P & & 0.93 \\
2 & \text{born in} & RG & W & \text{NYC} & C & & 0.96 \\
3 & \text{live in} & RG & W & \text{NYC} & C & & \\
4 & \text{grow up in} & RG & W & \text{Br} & C & & \\
5 & \text{live in} & RG & W & \text{Br} & C & & \\
6 & \text{grow up in} & RG & W & \text{Br} & C & & \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|}
\hline
R_1 & R_2 & C_1 & C_2 & w \\
\hline
\text{live in} & \text{born in} & W & P & 1.40 \\
\text{live in} & \text{born in} & W & C & 1.53 \\
\text{grow up in} & \text{born in} & W & P & 2.68 \\
\text{grow up in} & \text{born in} & W & C & 0.74 \\
\hline
\end{array}
\]
Grounding

\[
\text{SELECT} \ M1.R1 \ \text{AS} \ R, \\
T.x \ \text{AS} \ x, \ T.C1 \ \text{AS} \ C1, \\
T.y \ \text{AS} \ y, \ T.C2 \ \text{AS} \ C2 \\
\text{FROM} \ M1 \\
\text{JOIN} \ T \ \text{ON} \ M1.R2 = T.R \\
\text{AND} \ M1.C1 = T.C1 \ \text{AND} \ M1.C2 = T.C2;
\]

<table>
<thead>
<tr>
<th>$I$</th>
<th>$R$</th>
<th>$x$</th>
<th>$C_1$</th>
<th>$y$</th>
<th>$C_2$</th>
<th>$w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>born in</td>
<td>RG</td>
<td>$W$</td>
<td>Br</td>
<td>$P$</td>
<td>0.93</td>
</tr>
<tr>
<td>2</td>
<td>born in</td>
<td>RG</td>
<td>$W$</td>
<td>NYC</td>
<td>$C$</td>
<td>0.96</td>
</tr>
<tr>
<td>3</td>
<td>live in</td>
<td>RG</td>
<td>$W$</td>
<td>NYC</td>
<td>$C$</td>
<td>1.40</td>
</tr>
<tr>
<td>4</td>
<td>live in</td>
<td>RG</td>
<td>$W$</td>
<td>NYC</td>
<td>$C$</td>
<td>1.53</td>
</tr>
<tr>
<td>5</td>
<td>grow up in</td>
<td>RG</td>
<td>$W$</td>
<td>Br</td>
<td>$P$</td>
<td>2.68</td>
</tr>
<tr>
<td>6</td>
<td>grow up in</td>
<td>RG</td>
<td>$W$</td>
<td>Br</td>
<td>$P$</td>
<td>0.74</td>
</tr>
</tbody>
</table>

$M_1$

$\mathcal{T}$

<table>
<thead>
<tr>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>live in</td>
<td>born in</td>
<td>$W$</td>
<td>$P$</td>
<td>1.40</td>
</tr>
<tr>
<td>live in</td>
<td>born in</td>
<td>$W$</td>
<td>$C$</td>
<td>1.53</td>
</tr>
<tr>
<td>grow up in</td>
<td>born in</td>
<td>$W$</td>
<td>$P$</td>
<td>2.68</td>
</tr>
<tr>
<td>grow up in</td>
<td>born in</td>
<td>$W$</td>
<td>$C$</td>
<td>0.74</td>
</tr>
</tbody>
</table>
Grounding

\[
\begin{align*}
\text{SELECT} & \quad M3.R1 \quad \text{AS} \quad R, \\
& \quad T2.y \quad \text{AS} \quad x, \quad T2.C2 \quad \text{AS} \quad C1, \\
& \quad T3.y \quad \text{AS} \quad y, \quad T3.C2 \quad \text{AS} \quad C2 \\
\text{FROM} & \quad M3 \\
& \quad \text{JOIN} \quad T \quad \text{ON} \quad M3.R2 = T2.R \quad \text{AND} \\
& \quad \quad \quad \quad \quad \quad M3.C3 = T2.C1 \quad \text{AND} \quad M3.C1 = T2.C2 \\
& \quad \text{JOIN} \quad T \quad \text{ON} \quad M3.R3 = T3.R \quad \text{AND} \\
& \quad \quad \quad \quad \quad \quad M3.C3 = T3.C1 \quad \text{AND} \quad M3.C2 = T3.C2 \\
\text{WHERE} & \quad T2.x = T3.x;
\end{align*}
\]

\[\begin{array}{|c|c|c|c|c|}
\hline
I & R & x & C_1 & y & C_2 & w \\
\hline
1 & born in & RG & W & Br & P & 0.93 \\
2 & born in & RG & W & NYC & C & 0.96 \\
3 & live in & RG & W & NYC & C & \\
4 & grow up in & RG & W & NYC & C & \\
5 & live in & RG & W & Br & P & \\
6 & grow up in & RG & W & Br & P & \\
7 & located in & Br & P & NYC & C & \\
\hline
\end{array}\]

\[\begin{array}{|c|c|c|c|c|c|c|}
\hline
R_1 & R_2 & R_3 & C_1 & C_2 & C_3 & w \\
\hline
\text{located in} & \text{live in} & \text{born in} & P & C & W & 0.32 \\
\text{located in} & \text{born in} & \text{born in} & P & C & W & 0.52 \\
\hline
\end{array}\]
Grounding

\[
\begin{align*}
\text{SELECT} & \quad \text{M3.R1 AS R,} \\
& \quad \text{T2.y AS x, T2.C2 AS C1,} \\
& \quad \text{T3.y AS y, T3.C2 AS C2} \\
\text{FROM} & \quad \text{M3} \\
& \quad \text{JOIN T T2 ON M3.R2 = T2.R AND} \\
& \quad \quad \text{M3.C3 = T2.C1 AND M3.C1 = T2.C2} \\
& \quad \text{JOIN T T3 ON M3.R3 = T3.R AND} \\
& \quad \quad \text{M3.C3 = T3.C1 AND M3.C2 = T3.C2} \\
\text{WHERE} & \quad \text{T2.x = T3.x;}
\end{align*}
\]

\[\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
I & R & x & C_1 & y & C_2 & w \\
\hline
1 & born in & RG & W & Br & P & 0.93 \\
2 & born in & RG & W & NYC & C & 0.96 \\
3 & live in & RG & W & NYC & C & 0.96 \\
4 & grow up in & RG & W & Br & P & 0.93 \\
5 & live in & RG & W & Br & P & 0.93 \\
6 & grow up in & RG & W & Br & P & 0.93 \\
\hline
\end{array}\]

\[\begin{array}{|c|c|c|c|c|c|}
\hline
R_1 & R_2 & R_3 & C_1 & C_2 & C_3 & w \\
\hline
\text{located in} & \text{live in} & \text{live in} & P & C & W & 0.32 \\
\text{located in} & \text{born in} & \text{born in} & P & C & W & 0.52 \\
\hline
\end{array}\]
**Grounding**

```
FROM M3
WHERE T2.x = T3.x;
```

<table>
<thead>
<tr>
<th>$I$</th>
<th>$R$</th>
<th>$x$</th>
<th>$C_1$</th>
<th>$y$</th>
<th>$C_2$</th>
<th>$w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>born in</td>
<td>RG</td>
<td>$W$</td>
<td>Br</td>
<td>$P$</td>
<td>0.93</td>
</tr>
<tr>
<td>2</td>
<td>born in</td>
<td>RG</td>
<td>$W$</td>
<td>NYC</td>
<td>$C$</td>
<td>0.96</td>
</tr>
<tr>
<td>3</td>
<td>live in</td>
<td>RG</td>
<td>$W$</td>
<td>NYC</td>
<td>$C$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>grow up in</td>
<td>RG</td>
<td>$W$</td>
<td>NYC</td>
<td>$C$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>live in</td>
<td>RG</td>
<td>$W$</td>
<td>Br</td>
<td>$P$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>grow up in</td>
<td>RG</td>
<td>$W$</td>
<td>Br</td>
<td>$P$</td>
<td></td>
</tr>
</tbody>
</table>

$\mathcal{T}$

$M_3$

<table>
<thead>
<tr>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$R_3$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>located in</td>
<td>live in</td>
<td>live in</td>
<td>$P$</td>
<td>$C$</td>
<td>$W$</td>
<td>0.32</td>
</tr>
<tr>
<td>located in</td>
<td>born in</td>
<td>born in</td>
<td>$P$</td>
<td>$C$</td>
<td>$W$</td>
<td>0.52</td>
</tr>
</tbody>
</table>
Grounding

$$\begin{align*}
&\text{SELECT } M3.R1 \text{ AS } R, \\
&\quad T2.y \text{ AS } x, T2.C2 \text{ AS } C1, \\
&\quad T3.y \text{ AS } y, T3.C2 \text{ AS } C2 \\
&\text{FROM } M3 \\
&\quad \text{JOIN } T \text{ ON } M3.R2 = T2.R \text{ AND } M3.C3 = T2.C1 \text{ AND } M3.C1 = T2.C2 \\
&\quad \text{JOIN } T \text{ ON } M3.R3 = T3.R \text{ AND } M3.C3 = T3.C1 \text{ AND } M3.C2 = T3.C2 \\
&\text{WHERE } T2.x = T3.x;
\end{align*}$$

\begin{tabular}{|c|c|c|c|c|c|}
\hline
\textbf{I} & \textbf{R} & \textbf{x} & \textbf{C}_1 & \textbf{y} & \textbf{C}_2 & \textbf{w} \\
\hline
1 & born in & RG & W & Br & P & 0.93 \\
2 & born in & RG & W & NYC & C & 0.96 \\
3 & live in & RG & W & NYC & C & \\
4 & grow up in & RG & W & NYC & C & \\
5 & live in & RG & W & Br & P & \\
6 & grow up in & RG & W & Br & P & \\
7 & located in & Br & P & NYC & C & \\
\hline
\end{tabular}

\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
\textbf{R}_1 & \textbf{R}_2 & \textbf{R}_3 & \textbf{C}_1 & \textbf{C}_2 & \textbf{C}_3 & \textbf{w} \\
\hline
located in & live in & live in & P & C & W & 0.32 \\
located in & born in & born in & P & C & W & 0.52 \\
\hline
\end{tabular}
Grounding Efficiency
Sherlock-ReVerb KB

**TUFFY**  State-of-the-art MLN inference engine;
**ReVerb**  400K extracted facts from web text corpus;
**Sherlock**  31K inference rules learned from ReVerb.

| # relations | 82,768 |
| # rules     | 30,912 |
| # entities  | 277,216 |
| # facts     | 407,247 |

Table: Sherlock-ReVerb KB statistics
The ProbKB System

Grounding Efficiency
SHERLOCK-ReVERB KB

<table>
<thead>
<tr>
<th>Systems</th>
<th>Load</th>
<th>Round 1</th>
<th>Round 2</th>
<th>Round 3</th>
<th>Round 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProbKB-p</td>
<td>0.25</td>
<td>0.07</td>
<td>0.07</td>
<td>0.15</td>
<td>0.48</td>
</tr>
<tr>
<td>ProbKB</td>
<td>0.03</td>
<td>0.05</td>
<td>0.12</td>
<td>0.23</td>
<td>1.28</td>
</tr>
<tr>
<td>Tuffy-Τ</td>
<td>18.22</td>
<td>1.92</td>
<td>9.40</td>
<td>22.40</td>
<td>44.77</td>
</tr>
<tr>
<td># records</td>
<td>396K</td>
<td>420K</td>
<td>456K</td>
<td>580K</td>
<td>1.5M</td>
</tr>
</tbody>
</table>

Table: ReVERB-SHERLOCK case study
Grounding Efficiency
Synthetic KBs

(a) Varying # Rules: 311x-speedup
(b) Varying # Facts: 237x-speedup
(c) MPP Improvements: 6.3x-speedup
Quality Control
Inference Errors

- Incorrect extractions;
- Incorrect rules;
- Ambiguous entities;
- Propagated errors.
Quality Control

- Semantic constraints and ambiguity detection:

<table>
<thead>
<tr>
<th>Functional Relations</th>
<th>Violating Facts</th>
<th>Ambiguous Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>born in</td>
<td>born in(Mandel, Berlin)</td>
<td>Leonard Mandel</td>
</tr>
<tr>
<td>born in</td>
<td>born in(Mandel, New York City)</td>
<td>Johnny Mandel</td>
</tr>
<tr>
<td>born in</td>
<td>born in(Mandel, Chicago)</td>
<td>Tom Mandel (futurist)</td>
</tr>
<tr>
<td>grow up in</td>
<td>grow up in(Miller, Placentia)</td>
<td>Dustin Miller</td>
</tr>
<tr>
<td>grow up in</td>
<td>grow up in(Miller, New York City)</td>
<td>Alan Gifford Miller</td>
</tr>
<tr>
<td>grow up in</td>
<td>grow up in(Miller, New Orleans)</td>
<td>Taylor Miller</td>
</tr>
<tr>
<td>located in</td>
<td>located in(Regional office, Glasgow)</td>
<td>McCarthy &amp; Stone regional offices</td>
</tr>
<tr>
<td>located in</td>
<td>located in(Regional office, Panama City)</td>
<td>OCHA regional offices</td>
</tr>
<tr>
<td>located in</td>
<td>located in(Regional office, South Bend)</td>
<td>Indiana Landmarks regional offices</td>
</tr>
<tr>
<td>capital of</td>
<td>capital of(Delhi, India)</td>
<td>(Incorrect extraction)</td>
</tr>
<tr>
<td>capital of</td>
<td>capital of(Calcutta, India)</td>
<td></td>
</tr>
</tbody>
</table>

- Statistical Rule Cleaning:

\[ P(\text{Head}(\ldots)|\text{Body}(\ldots)) \gg P(\text{Head}(\ldots)) \]
Results

(a) 0.6 higher precision.

(b) Error sources.
Outline

1. Introduction
   - Knowledge Bases
   - Knowledge Expansion

2. The ProbKB System
   - Probabilistic Knowledge Bases
   - ProbKB Architecture
   - Grounding
   - Quality Control

3. Conclusion
   - Conclusion
We present ProbKB, a Probabilistic Knowledge Base system.

We design a novel relational model and an efficient SQL-based inference algorithm that applies inference rules \textit{in batches}.

We use MPP databases to parallelize the inference process.

We combine state-of-the-art data cleaning techniques to improve knowledge quality.

Future work will focus on rules and constraints learning.
Related Work

**Tuffy**: Scaling up Statistical Inference in Markov Logic Networks using an RDBMS
http://hazy.cs.wisc.edu/hazy/tuffy

**OpenIE**: Open Information Extraction
http://openie.cs.washington.edu

**Sherlock**: Learning First-Order Horn Clauses from Web Text
http://www.cs.washington.edu/research/sherlock-hornclauses

**Leibniz**: Identifying Functional Relations in Web Text
http://knowitall.cs.washington.edu/leibniz
Thank you!

- Yang Chen:  
  http://cise.ufl.edu/~yang

- Data Science Research at UF:  
  http://dsr.cise.ufl.edu

- Questions?