Mining Knowledge Graphs from Text

WSDM 2018

Jay Pujara, Sameer Singh
Tutorial Overview

https://kgtutorial.github.io

Part 1: Knowledge Graphs

Part 2: Knowledge Extraction

Part 3: Graph Construction

Part 4: Critical Analysis
Tutorial Outline

1. Knowledge Graph Primer [Jay]
2. Knowledge Extraction Primer [Jay]
3. Knowledge Graph Construction
   a. Probabilistic Models [Jay]
   b. Embedding Techniques [Sameer]
4. Critical Overview and Conclusion [Sameer]
Knowledge Graph Construction

TOPICS:

Problem Setting
Probabilistic Models
Embedding Techniques
Knowledge Graph Construction

TOPICS:

PROBLEM SETTING

PROBABILISTIC MODELS

EMBEDDING TECHNIQUES
Reminder: Basic problems

- **Who** are the entities (nodes) in the graph?
- **What** are their attributes and types (labels)?
- **How** are they related (edges)?
Graph Construction Issues

Extracted knowledge is:

- ambiguous:
  - Ex: Beetles, beetles, Beatles
  - Ex: citizenOf, livedIn, bornIn
Graph Construction Issues

Extracted knowledge is:

• ambiguous

• incomplete
  ◦ Ex: missing relationships
  ◦ Ex: missing labels
  ◦ Ex: missing entities
Graph Construction Issues

Extracted knowledge is:

• ambiguous

• incomplete

• inconsistent
  ◦ Ex: Cynthia Lennon, Yoko Ono
  ◦ Ex: exclusive labels (alive, dead)
  ◦ Ex: domain-range constraints
Graph Construction Issues

Extracted knowledge is:

- ambiguous
- incomplete
- inconsistent
Graph Construction approach

- Graph construction **cleans** and **completes** extraction graph

- Incorporate ontological constraints and relational patterns

- Discover statistical relationships within knowledge graph
Knowledge Graph Construction

TOPICS:

Problem Setting

Probabilistic Models

Embedding Techniques
Graph Construction
Probabilistic Models

TOPICS:

OVERVIEW

GRAPHICAL MODELS

RANDOM WALK METHODS
Graph Construction
Probabilistic Models

TOPICS:

OVERVIEW

GRAPHICAL MODELS

RANDOM WALK METHODS
Beyond Pure Reasoning

- Classical AI approach to knowledge: reasoning

\[ \text{Lbl}(\text{Socrates, Man}) \land \text{Sub}(\text{Man, Mortal}) \rightarrow \text{Lbl}(\text{Socrates, Mortal}) \]
Beyond Pure Reasoning

• Classical AI approach to knowledge: reasoning

  Lbl(Socrates, Man) & Sub(Man, Mortal) -> Lbl(Socrates, Mortal)

• Reasoning difficult when extracted knowledge has errors
Beyond Pure Reasoning

• Classical AI approach to knowledge: reasoning
  Lbl(Socrates, Man) & Sub(Man, Mortal) -> Lbl(Socrates, Mortal)
• Reasoning difficult when extracted knowledge has errors
• Solution: probabilistic models
  P(Lbl(Socrates, Mortal) | Lbl(Socrates, Man) = 0.9)
Graph Construction
Probabilistic Models

TOPICS:

OVERVIEW

GRAPHICAL MODELS

RANDOM WALK METHODS
Graphical Models: Overview

• Define **joint probability distribution** on knowledge graphs

• Each candidate fact in the knowledge graph is a **variable**

• Statistical signals, ontological knowledge and rules parameterize the **dependencies** between variables

• Find most likely knowledge graph by **optimization/sampling**
Knowledge Graph Identification

Define a graphical model to perform all three of these tasks simultaneously!

• **Who** are the entities (nodes) in the graph?

• **What** are their attributes and types (labels)?

• **How** are they related (edges)?
Knowledge Graph Identification

\[ P(\text{Who, What, How} \mid \text{Extractions}) \]
Probabilistic Models

- Use dependencies between facts in KG

- Probability defined *jointly* over facts
What determines probability?

- Statistical signals from text extractors and classifiers
What determines probability?

- **Statistical signals from text extractors and classifiers**
  - \( P(\text{R(John, Spouse, Yoko)}) = 0.75; \ P(\text{R(John, Spouse, Cynthia)}) = 0.25 \)
  - \( \text{LevenshteinSimilarity(Beatles, Beetles)} = 0.9 \)
What determines probability?

• Statistical signals from text extractors and classifiers

• Ontological knowledge about domain
What determines probability?

• Statistical signals from text extractors and classifiers

• Ontological knowledge about domain
  • Functional(Spouse) & R(A,Spouse,B) -> !R(A,Spouse,C)
  • Range(Spouse, Person) & R(A,Spouse,B) -> Type(B, Person)
What determines probability?

- Statistical signals from text extractors and classifiers
- Ontological knowledge about domain
- Rules and patterns mined from data
What determines probability?

• Statistical signals from text extractors and classifiers

• Ontological knowledge about domain

• Rules and patterns mined from data
  • R(A, Spouse, B) & R(A, Lives, L) -> R(B, Lives, L)
  • R(A, Spouse, B) & R(A, Child, C) -> R(B, Child, C)
What determines probability?

- **Statistical signals from text extractors and classifiers**
  - \( P(R(John, \text{Spouse}, Yoko)) = 0.75; \ P(R(John, \text{Spouse}, Cynthia)) = 0.25 \)
  - LevenshteinSimilarity(Beatles, Beetles) = 0.9

- **Ontological knowledge about domain**
  - Functional(\text{Spouse}) & R(A, \text{Spouse}, B) -> !R(A, \text{Spouse}, C)
  - Range(\text{Spouse, Person}) & R(A, \text{Spouse}, B) -> Type(B, \text{Person})

- **Rules and patterns mined from data**
  - R(A, \text{Spouse}, B) & R(A, \text{Lives}, L) -> R(B, \text{Lives}, L)
  - R(A, \text{Spouse}, B) & R(A, \text{Child}, C) -> R(B, \text{Child}, C)
Example: The Fab Four

THE

BEATLES

SUPER HEROES

THE FAB 4

AMAZING NEW SUPER HEROES OF SCIENCE

ELL 2c

THE LADY

THEY GET
Illustration of KG Identification

Uncertain Extractions:
.5: Lbl(Fab Four, novel)
.7: Lbl(Fab Four, musician)
.9: Lbl(Beatles, musician)
.8: Rel(Beatles, AlbumArtist, Abbey Road)
Illustration of KG Identification

**Uncertain Extractions:**
- .5: Lbl(Fab Four, novel)
- .7: Lbl(Fab Four, musician)
- .9: Lbl(Beatles, musician)
- .8: Rel(Beatles, AlbumArtist, Abbey Road)
Illustration of KG Identification

Uncertain Extractions:
.5: Lbl(Fab Four, novel)
.7: Lbl(Fab Four, musician)
.9: Lbl(Beatles, musician)
.8: Rel(Beatles, AlbumArtist, Abbey Road)

Ontology:
Dom(albumArtist, musician)
Mut(novel, musician)
Illustration of KG Identification

**Uncertain Extractions:**
.5: Lbl(Fab Four, novel)
.7: Lbl(Fab Four, musician)
.9: Lbl(Beatles, musician)
.8: Rel(Beatles, AlbumArtist, Abbey Road)

**Ontology:**
Dom(albumArtist, musician)
Mut(novel, musician)

**Entity Resolution:**
SameEnt(Fab Four, Beatles)
Illustration of KG Identification

Uncertain Extractions:
.5: Lbl(Fab Four, novel)
.7: Lbl(Fab Four, musician)
.9: Lbl(Beatles, musician)
.8: Rel(Beatles, AlbumArtist, Abbey Road)

Ontology:
Dom(albumArtist, musician)
Mut(novel, musician)

Entity Resolution:
SameEnt(Fab Four, Beatles)

After Knowledge Graph Identification

PUJARA+ISWC13; PUJARA+AIMAG15
Probabilistic graphical model for KG

Lbl(Beatles, novel)

Rel(Beatles, AlbumArtist, Abbey Road)

Lbl(Beatles, musician)

Lbl(Fab Four, novel)

Rel(Fab Four, AlbumArtist, Abbey Road)

Lbl(Fab Four, musician)
Defining graphical models

- Many options for defining a graphical model

- We focus on two approaches, MLNs and PSL, that use rules

  - **MLNs** treat facts as Boolean, use sampling for satisfaction

  - **PSL** infers a “truth value” for each fact via optimization
## Rules for KG Model

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>100</strong>: Subsumes($L_1, L_2$) &amp; Label($E, L_1$)</td>
<td>$\rightarrow$ Label($E, L_2$)</td>
</tr>
<tr>
<td><strong>100</strong>: Exclusive($L_1, L_2$) &amp; Label($E, L_1$)</td>
<td>$\rightarrow$ !Label($E, L_2$)</td>
</tr>
<tr>
<td><strong>100</strong>: Inverse($R_1, R_2$) &amp; Relation($R_1, E, O$)</td>
<td>$\rightarrow$ Relation($R_2, O, E$)</td>
</tr>
<tr>
<td><strong>100</strong>: Subsumes($R_1, R_2$) &amp; Relation($R_1, E, O$)</td>
<td>$\rightarrow$ Relation($R_2, E, O$)</td>
</tr>
<tr>
<td><strong>100</strong>: Exclusive($R_1, R_2$) &amp; Relation($R_1, E, O$)</td>
<td>$\rightarrow$ !Relation($R_2, E, O$)</td>
</tr>
<tr>
<td><strong>100</strong>: Domain($R, L$) &amp; Relation($R, E, O$)</td>
<td>$\rightarrow$ Label($E, L$)</td>
</tr>
<tr>
<td><strong>100</strong>: Range($R, L$) &amp; Relation($R, E, O$)</td>
<td>$\rightarrow$ Label($O, L$)</td>
</tr>
<tr>
<td><strong>10</strong>: SameEntity($E_1, E_2$) &amp; Label($E_1, L$)</td>
<td>$\rightarrow$ Label($E_2, L$)</td>
</tr>
<tr>
<td><strong>10</strong>: SameEntity($E_1, E_2$) &amp; Relation($R, E_1, O$)</td>
<td>$\rightarrow$ Relation($R, E_2, O$)</td>
</tr>
<tr>
<td><strong>1</strong>: Label_OBIE($E, L$)</td>
<td>$\rightarrow$ Label($E, L$)</td>
</tr>
<tr>
<td><strong>1</strong>: Label_OpenIE($E, L$)</td>
<td>$\rightarrow$ Label($E, L$)</td>
</tr>
<tr>
<td><strong>1</strong>: Relation_Pattern($R, E, O$)</td>
<td>$\rightarrow$ Relation($R, E, O$)</td>
</tr>
<tr>
<td><strong>1</strong>:</td>
<td>$\rightarrow$ !Relation($R, E, O$)</td>
</tr>
<tr>
<td><strong>1</strong>:</td>
<td>$\rightarrow$ !Label($E, L$)</td>
</tr>
</tbody>
</table>
Rules to Distributions

- Rules are *grounded* by substituting literals into formulas

\[ w_r : \text{SAMEENT}(\text{Fab Four}, \text{Beatles}) \land \text{LBL}(\text{Beatles, musician}) \Rightarrow \text{LBL}(\text{Fab Four, musician}) \]

- Each ground rule has a *weighted satisfaction* derived from the formula’s truth value

\[ P(G|E) = \frac{1}{Z} \exp \left[ \sum_{r \in R} w_r \phi_r(G, E) \right] \]

- Together, the ground rules provide a joint probability distribution over knowledge graph facts, conditioned on the extractions

JIANG+ICDM12; PUJARA+ISWC13
Probability Distribution over KGs

\[ P(G \mid E) = \frac{1}{Z} \exp \left[ - \sum_{r \in R} w_r \varphi_r(G) \right] \]

\[ \text{CandLbl}_T(\text{FabFour}, \text{novel}) \quad \Rightarrow \quad \text{Lbl}(\text{FabFour}, \text{novel}) \]

\[ \text{Mut}(\text{novel}, \text{musician}) \quad \land \quad \text{Lbl}(\text{Beatles}, \text{novel}) \quad \Rightarrow \quad \neg \text{Lbl}(\text{Beatles}, \text{musician}) \]

\[ \text{SameEnt}(\text{Beatles}, \text{FabFour}) \quad \land \quad \text{Lbl}(\text{Beatles}, \text{musician}) \quad \Rightarrow \quad \text{Lbl}(\text{FabFour}, \text{musician}) \]
\[\phi_1\] \text{CANDLBL}_{\text{struct}}(\text{FabFour, novel}) \\
\Rightarrow \quad \text{LBL}(\text{FabFour, novel})

\[\phi_2\] \text{CANDREL}_{\text{pat}}(\text{Beatles, AlbumArtist, AbbeyRoad}) \\
\Rightarrow \quad \text{REL}(\text{Beatles, AlbumArtist, AbbeyRoad})

\[\phi_3\] \text{SAMEENT}(\text{Beatles, FabFour}) \\
\wedge \quad \text{LBL}(\text{Beatles, musician}) \\
\Rightarrow \quad \text{LBL}(\text{FabFour, musician})

\[\phi_4\] \text{DOM}(\text{AlbumArtist, musician}) \\
\wedge \quad \text{REL}(\text{Beatles, AlbumArtist, AbbeyRoad}) \\
\Rightarrow \quad \text{LBL}(\text{Beatles, musician})

\[\phi_5\] \text{MUT}(\text{musician, novel}) \\
\wedge \quad \text{LBL}(\text{FabFour, musician}) \\
\Rightarrow \quad \neg \text{LBL}(\text{FabFour, novel})
How do we get a knowledge graph?

Have: $P(KG)$ for all KGs

Need: best KG

MAP inference: optimizing over distribution to find the best knowledge graph
Inference and KG optimization

• Finding the best KG satisfying weighed rules: NP Hard

• MLNs [discrete]: Monte Carlo sampling methods
  • Solution quality dependent on burn-in time, iterations, etc.

• PSL [continuous]: optimize convex linear surrogate
  • Fast optimization, ¾-optimal MAX SAT lower bound
Graphical Models Experiments

Data: ~1.5M extractions, ~70K ontological relations, ~500 relation/label types
Task: Collectively construct a KG and evaluate on 25K target facts

Comparisons:
Extract: Average confidences of extractors for each fact in the NELL candidates
Rules: Default, rule-based heuristic strategy used by the NELL project
MLN: Jiang+, ICDM12 – estimates marginal probabilities with MC-SAT
PSL: Pujara+, ISWC13 – convex optimization of continuous truth values with ADMM

Running Time: Inference completes in 10 seconds, values for 25K facts

<table>
<thead>
<tr>
<th></th>
<th>AUC</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract</td>
<td>.873</td>
<td>.828</td>
</tr>
<tr>
<td>Rules</td>
<td>.765</td>
<td>.673</td>
</tr>
<tr>
<td>MLN (Jiang, 12)</td>
<td>.899</td>
<td>.836</td>
</tr>
<tr>
<td>PSL (Pujara, 13)</td>
<td>.904</td>
<td>.853</td>
</tr>
</tbody>
</table>

JIANG+ICDM12; PUJARA+ISWC13
Graphical Models: Pros/Cons

**BENEFITS**
- Define probability distribution over KGs
- Easily specified via rules
- Fuse knowledge from many different sources

**DRAWBACKS**
- Requires optimization over all KG facts - overkill
- Dependent on rules from ontology/expert
- Require probabilistic semantics - unavailable
Graph Construction
Probabilistic Models

TOPICS:

OVERVIEW

GRAPHICAL MODELS

RANDOM WALK METHODS
Random Walk Overview

- Given: a query of an **entity** and **relation**

- Starting at the entity, **randomly walk** the KG

- Random walk ends when reaching an appropriate **goal**

- Learned **parameters** bias choices in the random walk

- Output **relative probabilities** of goal states
Random Walk Illustration

Query: R(Lennon, PlaysInstrument, ?)
Query: \( R(\text{Lennon, PlaysInstrument, ?}) \)
Random Walk Illustration

Query: $R(\text{Lennon, PlaysInstrument, ?})$
Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)
Random Walk Illustration

Query Q: \( R(\text{Lennon, PlaysInstrument, ?}) \)
Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)
Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)

Path

\[ P(Q|\pi=\text{coworker,playsInstrument}) W_\pi \]
Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)

\[ P(Q | \pi = \text{coworker, playsInstrument}) \] \[ W_\pi \]
Random Walk Illustration

Query Q: \( R(\text{Lennon, PlaysInstrument, ?}) \)

\[ P(Q|\pi=\langle \text{coworker, playsInstrument} \rangle) W_{\pi} \]
Random Walk Illustration

Query Q: \( R(\text{Lennon, PlaysInstrument, } ?) \)
Random Walk Illustration

Query Q: $R(\text{Lennon, PlaysInstrument}, \, ?)$

$P(Q | \pi = \text{<albumArtist,hasInstrument>}) \cdot W_\pi$
Random Walk Illustration

Query Q: \( R(\text{Lennon}, \text{PlaysInstrument}, \, ?) \)

\[
P(Q|\pi=\langle \text{albumArtist}, \text{hasInstrument} \rangle) \cdot W_\pi
\]
Random Walk Illustration

Query: R(Lennon, PlaysInstrument, ?)
Recent Random Walk Methods

**PRA: Path Ranking Algorithm**
- Performs random walk of *imperfect knowledge graph*
- Estimates *transition probabilities* using KG
- For each relation, learns *parameters for paths* through the KG

**ProPPR: Programming with Personalized PageRank**
- Constructs *proof graph*
  - Nodes are partially-ground clauses with one or more facts
  - Edges are proof-transformations
- *Parameters* are learned for each *ground entity* and *rule*
Recent Random Walk Methods

PRA: Path Ranking Algorithm

- Performs random walk of imperfect knowledge graph
- Estimates transition probabilities using KG
- For each relation, learns parameters for paths through the KG

ProPPR: Programming with Personalized PageRank

- Constructs proof graph
  - Nodes are partially-ground clauses with one or more facts
  - Edges are proof-transformations
- Parameters are learned for each ground entity and rule
PRA in a nutshell

$$\text{score}(q.s \rightarrow e; q) = \sum_{\pi_i \in \Pi_b} P(q.s \rightarrow e; \pi_i) W_{\pi_i}$$
PRA in a nutshell

\[
\text{score}(q.s \rightarrow e; q) = \sum_{\pi_i \in \Pi_b} P(q.s \rightarrow e; \pi_i) W_{\pi_i}
\]

Filter paths based on HITS and accuracy
PRA in a nutshell

\[
\text{score}(q.s \rightarrow e; q) = \sum_{\pi_i \in \Pi_b} P(q.s \rightarrow e; \pi_i) W_{\pi_i}
\]

Filter paths based on HITS and accuracy

Estimate probabilities efficiently with dynamic programming
PRA in a nutshell

\[
\text{score}(q.s \rightarrow e; q) = \sum_{\pi_i \in \Pi_b} P(q.s \rightarrow e; \pi_i) W_{\pi_i}
\]

Filter paths based on HITS and accuracy

Estimate probabilities efficiently with dynamic programming

Path weights are learned with logistic regression
Recent Random Walk Methods

PRA: Path Ranking Algorithm

- Performs random walk of imperfect knowledge graph
- Estimates transition probabilities using KG
- For each relation, learns parameters for paths through the KG

ProPPR: ProbLog + Personalized PageRank

- Constructs proof graph
  - Nodes are partially-ground clauses with one or more facts
  - Edges are proof-transformations
- Parameters are learned for each ground entity and rule
ProPPR-ized PRA example

Query Q: R(Lennon, PlaysInstrument, ?)

R(X,PlaysInstrument,Y)
R(Coworker,X)
R(AlbumArtist,J)
R(HasInstrument,K)

Unbound variables in proof tree!
ProPPR-ized PRA example

Query Q: \( R(\text{Lennon}, \text{PlaysInstrument}, ?) \)

- \( R(X, \text{Coworker}, \text{X}) \)
- \( R(\text{X}, \text{PlaysInstrument}, \text{Y}) \)
- \( R(\text{J}, \text{AlbumArtist}, \text{J}) \)
- \( R(\text{J}, \text{HasInstrument}, \text{K}) \)

\( R(\text{L}, \text{Coworker}, \text{L}) \)
\( R(\text{L}, \text{PlaysInstrument}, \text{Y}) \)
Query Q: $R(\text{Lennon, PlaysInstrument, } ?)$

$R(\text{Lennon, Coworker, } X)$
$R(X, \text{PlaysInstrument, } Y)$

$R(\text{X, Coworker, } X)$
$R(\text{X, PlaysInstrument, } Y)$

$R(\text{Y, AlbumArtist, } J)$
$R(J, \text{HasInstrument, } K)$
ProPPR-ized PRA example

Query Q: R(Lennon, PlaysInstrument, ?)

R(,Coworker,X)
R(X,PlaysInstrument,Y)
R(,Coworker,)
R(,PlaysInstrument,Y)

R(,AlbumArtist,J)
R(J,HasInstrument,K)
ProPPR-ized PRA example

Query Q: \( R(\text{Lennon, PlaysInstrument, } ?) \)

\[
\begin{align*}
R(\text{Lennon, Coworker, } X) & \quad R(X, \text{PlaysInstrument, } Y) \\
R(\text{X, PlaysInstrument, } Y) & \quad R(\text{J, AlbumArtist, } K) \\
R(\text{J, HasInstrument, } K) & \quad R(\text{Y, PlaysInstrument}) \\
R(\text{Y, PlaysInstrument}) & \quad R(\text{Coworker}) \\
R(\text{Coworker}) & \quad R(\text{Coworker})
\end{align*}
\]
ProPPR-ized PRA example

Query Q: $R($Lennon, PlaysInstrument, ?$)

$$R($Lennon, Coworker, X$)$$
$$R($X, PlaysInstrument, Y$)$$

$$R($X, Coworker, Y$)$$
$$R($J, HasInstrument, K$)$$

$$R($J, AlbumArtist, ?$)$$
$$R($K, AlbumArtist, ?$)$$

$$R($K, HasInstrument, ?$)$$
ProPPR-ized PRA example

Query Q: \( R(\text{Lennon}, \text{PlaysInstrument}, \ ?) \)

\[
\begin{align*}
R(\text{Lennon}, \text{Coworker}, X) \\
R(X, \text{PlaysInstrument}, Y) \\
R(\text{Lennon}, \text{Coworker}, X) \\
R(X, \text{PlaysInstrument}, Y) \\
R(\text{Lennon}, \text{Coworker}, X) \\
R(X, \text{PlaysInstrument}, Y) \\
R(\text{AlbumArtist}, J) \\
R(J, \text{HasInstrument}, K) \\
R(\text{AlbumArtist}, J) \\
R(J, \text{HasInstrument}, K) \\
R(\text{AlbumArtist}, J) \\
R(J, \text{HasInstrument}, K)
\end{align*}
\]
ProPPR-ized PRA example

Query Q: R(Lennon, PlaysInstrument, ?)
ProPPR in a nutshell

\[
\min_w - \left( \sum_{k \in +} \log p_{\nu_0}[u^k_+] + \sum_{k \in -} \log(1 - p_{\nu_0}[u^k_-]) \right) + \mu \|w\|_2^2
\]

- Input: queries, positive answers, negative answers
- Goal: \( p_{\nu_0}[u^k_+] \geq p_{\nu_0}[u^k_-] \) (page rank from RW)
- Learn: random walk weights
- Train via stochastic gradient descent

WANG+MLJ15
Results from PRA and ProPPR

- Task:
  - 1M extractions for 3 domains;
  - ~100s of training queries
  - ~1000s of test queries
  - AUC of extractions alone is 0.7
Random Walks: Pros/Cons

**BENEFITS**

- KG query estimation independent of KG size
- Model training produces interpretable, logical rules
- Robust to noisy extractions through probabilistic form

**DRAWBACKS**

- Full KG completion task inefficient
- Training data difficult to obtain at scale
- Input must follow probabilistic semantics
Two classes of Probabilistic Models

<table>
<thead>
<tr>
<th>GRAPHICAL MODELS</th>
<th>RANDOM WALK METHODS</th>
</tr>
</thead>
<tbody>
<tr>
<td>◦ Possible facts in KG are variables</td>
<td>◦ Possible facts posed as queries</td>
</tr>
<tr>
<td>◦ Logical rules relate facts</td>
<td>◦ Random walks of the KG constitute “proofs”</td>
</tr>
<tr>
<td>◦ Probability $\propto$ satisfied rules</td>
<td>◦ Probability $\propto$ path lengths/transitions</td>
</tr>
<tr>
<td>◦ Universally-quantified</td>
<td>◦ Locally grounded</td>
</tr>
</tbody>
</table>