Knowledge Graph Tutorial

Jay Pujara
jaypujara.org
jay@cs.umd.edu
@jay_mlr

Sameer Singh
sameersingh.org
sameer@uci.edu
@sameer_

Knowledge Graph Primer

TOPICS:

- What is a Knowledge Graph?
- Why are Knowledge Graphs Important?
- Where do Knowledge Graphs come from?
- Knowledge Representation Choices
- Problem Overview

Knowledge Graph Primer

TOPICS:

WHAT IS A KNOWLEDGE GRAPH?

WHY ARE KNOWLEDGE GRAPHS IMPORTANT?

WHERE DO KNOWLEDGE GRAPHS COME FROM?

KNOWLEDGE REPRESENTATION CHOICES

PROBLEM OVERVIEW

Knowledge Graph

Essentially, KG is a sematic network, which models **the entities (including properties) and the relation between each other.**

What is a knowledge graph?

What is a knowledge graph?

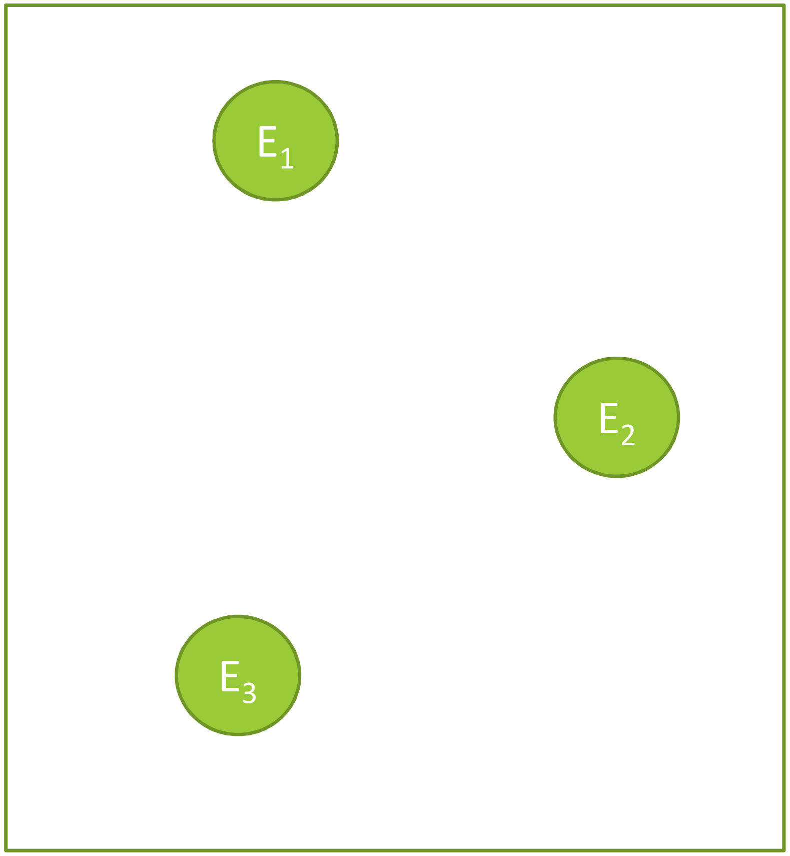
- Knowledge in graph form!

What is a knowledge graph?

- Knowledge in graph form!
- Captures entities, attributes, and relationships

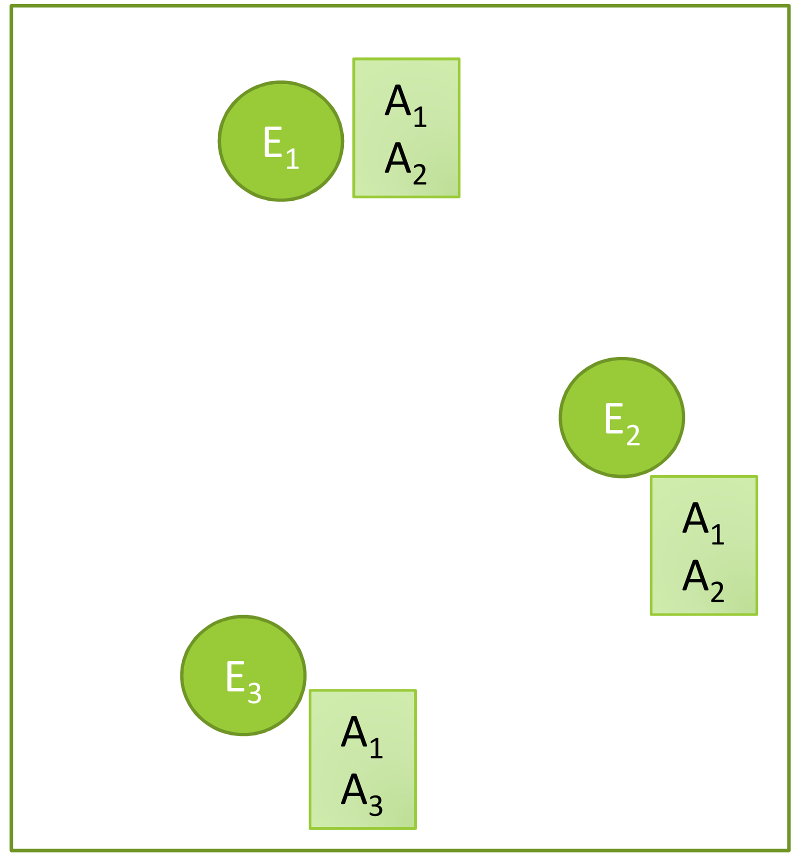
What is a knowledge graph?

- Knowledge in graph form!
- Captures entities, attributes, and relationships
- Nodes are entities



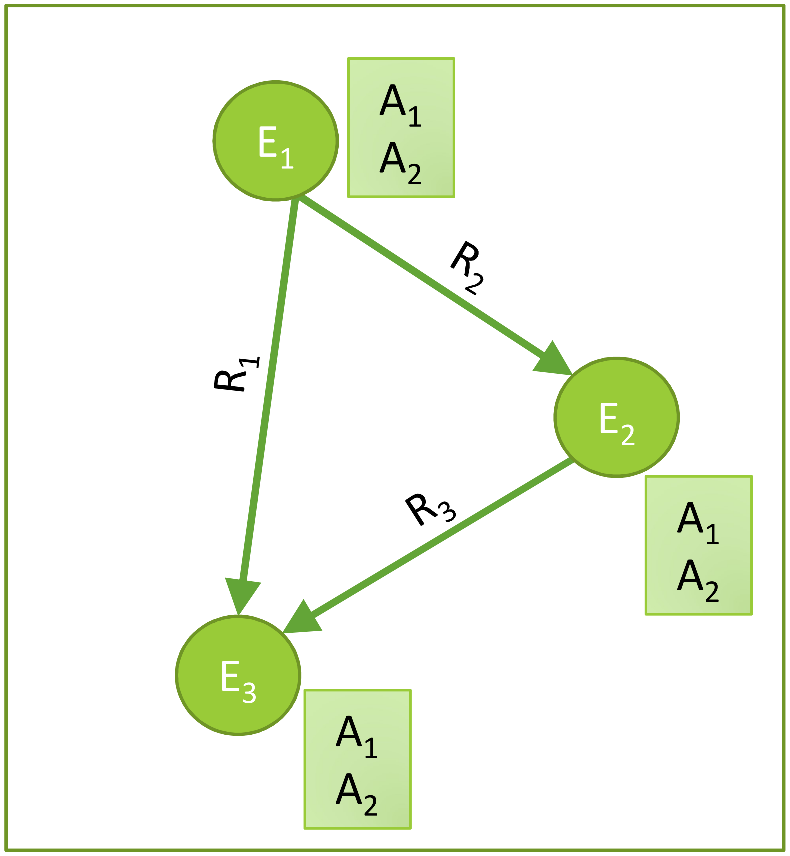
What is a knowledge graph?

- Knowledge in graph form!
- Captures entities, attributes, and relationships
- Nodes are entities
- Nodes are labeled with attributes (e.g., types)



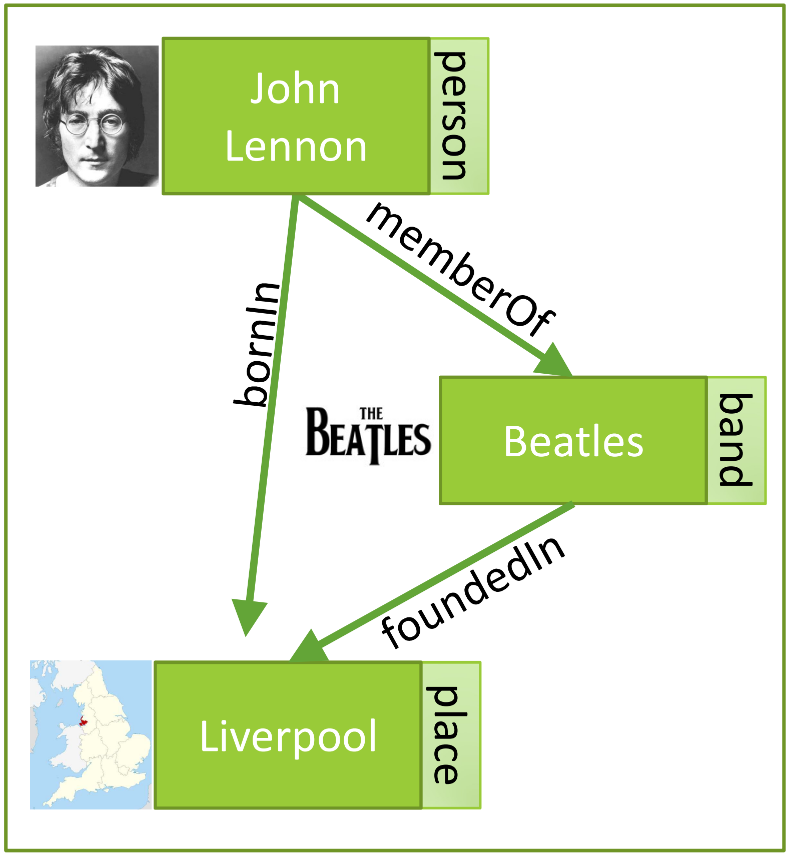
What is a knowledge graph?

- Knowledge in graph form!
- Captures entities, attributes, and relationships
- Nodes are entities
- Nodes are labeled with attributes (e.g., types)
- Typed edges between two nodes capture a relationship between entities



Example knowledge graph

- Knowledge in graph form!
- Captures entities, attributes, and relationships
- Nodes are entities
- Nodes are labeled with attributes (e.g., types)
- Typed edges between two nodes capture a relationship between entities



Knowledge Graph Primer

TOPICS:

- What is a Knowledge Graph?
- Why are Knowledge Graphs Important?
- Where do Knowledge Graphs come from?
- Knowledge Representation Choices
- Problem Overview

Why knowledge graphs?

- Humans:
  - Combat information overload
  - Explore via intuitive structure
  - Tool for supporting knowledge-driven tasks

- AIs:
  - Key ingredient for many AI tasks
  - Bridge from data to human semantics
  - Use decades of work on graph analysis

Interdisciplinary Research

Database

RDF Database

Data Integration 、 Knowledge Fusion

Natural Language Processing

Information Extraction

Semantic Parsing

Machine Learning

Knowledge Representation (Graph Embedding)

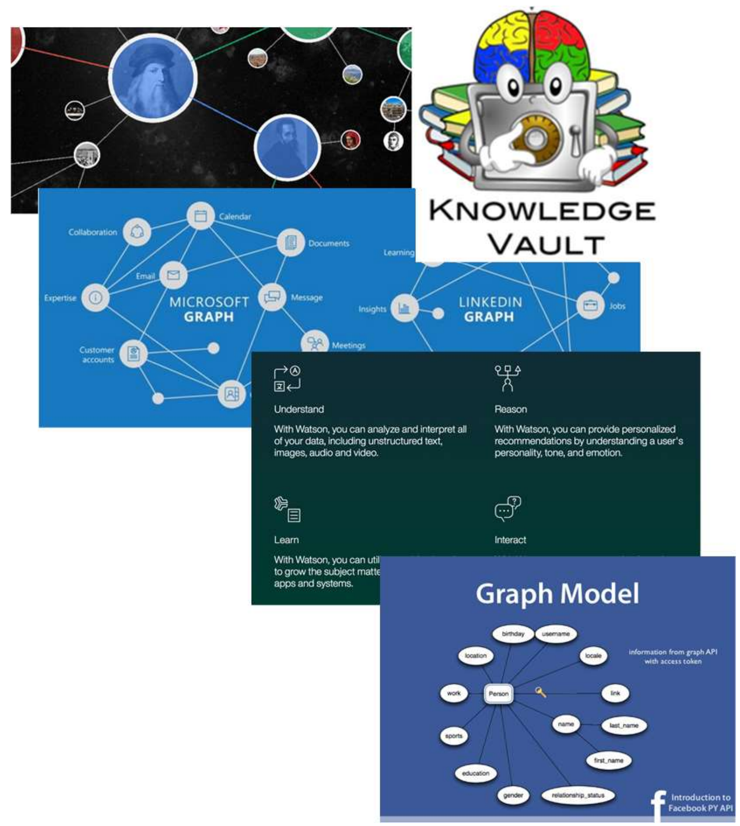
Knowledge Engineering

KB construction

Rule-based Reasoning

Knowledge Graphs & Industry

- Google Knowledge Graph
  - Google Knowledge Vault
- Amazon Product Graph
- Facebook Graph API
- IBM Watson
- Microsoft Satori
  - Project Hanover/Literome
- LinkedIn Knowledge Graph
- Yandex Object Answer
- Diffbot, GraphIQ, Maana, ParseHub, Reactor Labs, SpazioDati



Knowledge Graph Primer

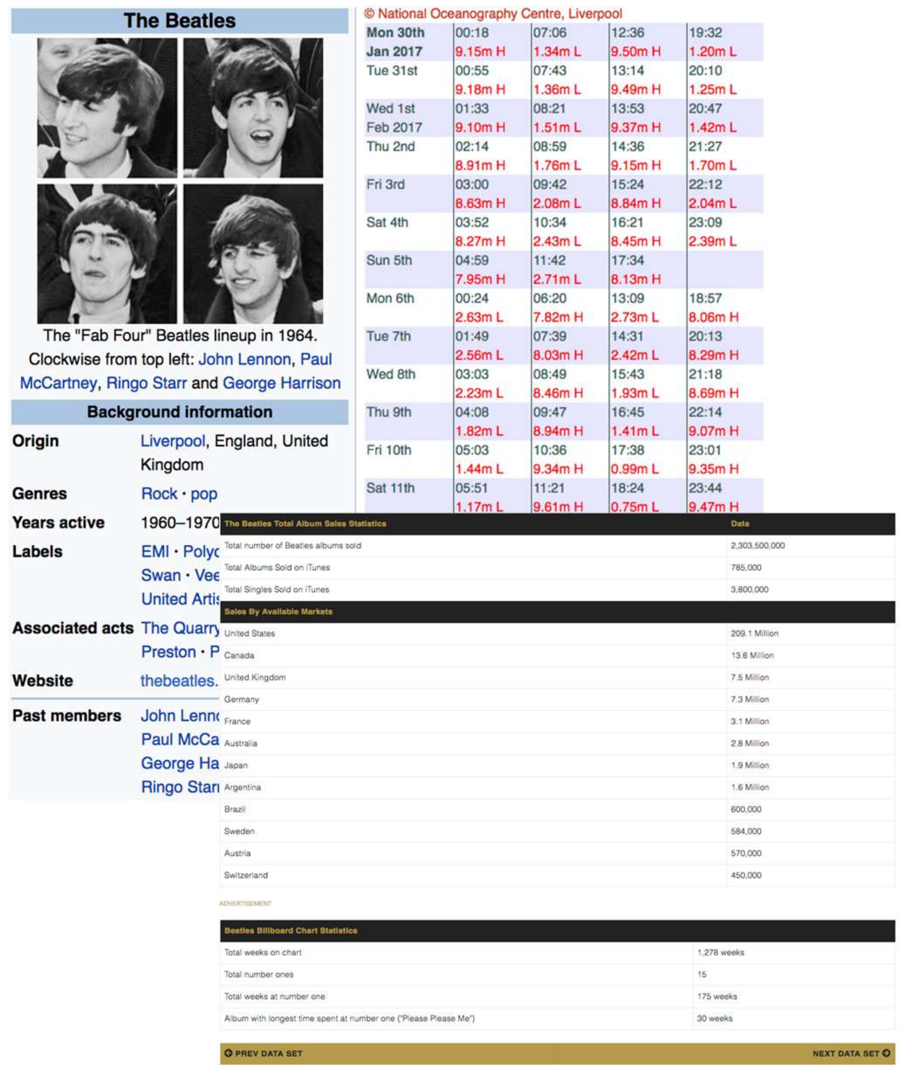
TOPICS:

- What is a Knowledge Graph?
- Why are Knowledge Graphs Important?
- Where do Knowledge Graphs come from?
- Knowledge Representation Choices
- Problem Overview

Where do knowledge graphs come from?

Where do knowledge graphs come from?

- Structured Text
  - Wikipedia Infoboxes, tables, databases, social nets



Where do knowledge graphs come from?

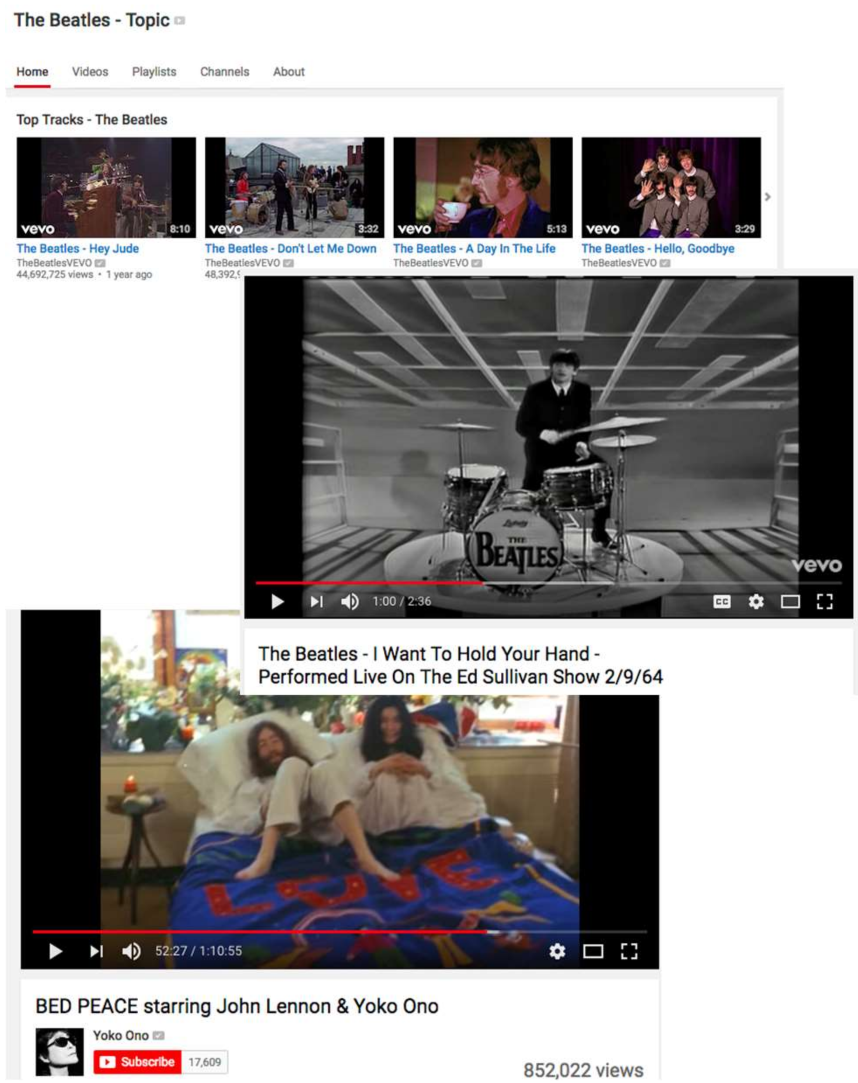
- Structured Text
  - Wikipedia Infoboxes, tables, databases, social nets
- Unstructured Text
  - WWW, news, social media, reference articles

Where do knowledge graphs come from?

- Structured Text
  - Wikipedia Infoboxes, tables, databases, social nets
- Unstructured Text
  - WWW, news, social media, reference articles
- Images

Where do knowledge graphs come from?

- Structured Text
    - Wikipedia Infoboxes, tables, databases, social nets
- Unstructured Text
    - WWW, news, social media, reference articles
- Images
- Video
    - YouTube, video feeds



Knowledge Representation

- Decades of research into knowledge representation

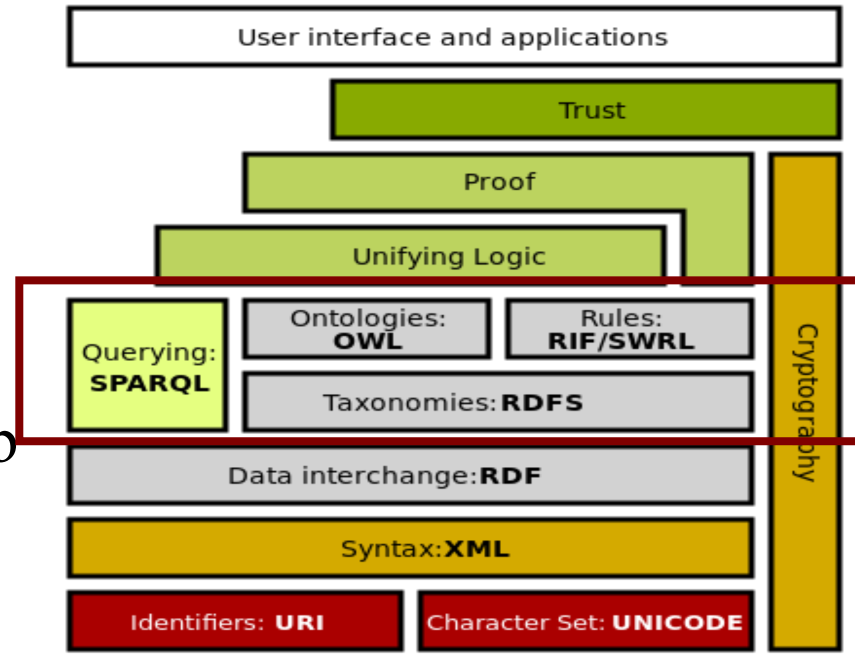
- Most knowledge graph implementations use RDF triples
  - <rdf:subject, rdf:predicate, rdf:object> : r(s,p,o)
  - Temporal scoping, reification, and skolemization...

- ABox (assertions) versus TBox (terminology)

- Common ontological primitives
  - rdfs:domain, rdfs:range, rdf:type, rdfs:subClassOf, rdfs:subPropertyOf, ...
  - owl:inverseOf, owl:TransitiveProperty, owl:FunctionalProperty, ...

Resource Description Framework (RDF)

- RDF is an **de facto standard** for Knowledge Graph (KG).
- RDF is a **language** for the conceptual modeling of information about web resources
- A **building block** of semantic web
- Make the information on the web and the interrelationships among them "**Machine Understandable**"



RDF and Semantic Web

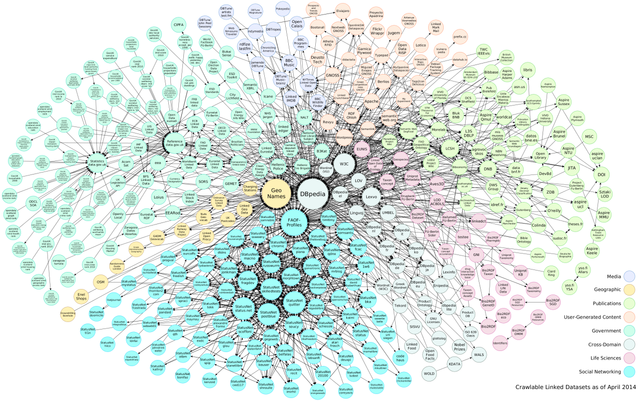
- RDF is a language for the conceptual modeling of information about web resources
- A building block of semantic web
  - Facilitates exchange of information
  - Search engines can retrieve more relevant information
  - Facilitates data integration (mashes)
- Machine understandable
  - Understand the information on the web and the interrelationships among them

RDF Uses

- Yago and DBPedia extract facts from Wikipedia & represent as RDF $\rightarrow$ structural queries
- Communities build RDF data
  - E.g., biologists: Bio2RDF and Uniprot RDF
- Web data integration
  - Linked Data Cloud
- ...

RDF Data Volumes . . .

- . . . are growing – and fast
  - Linked data cloud currently consists of 325 datasets with >25B triples
  - Size almost doubling every year



April '14:
1091 datasets, ???
triples

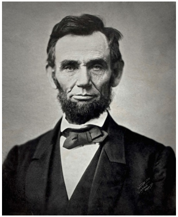
Max Schmachtenberg, Christian Bizer, and Heiko Paulheim: Adoption of Linked Data Best Practices in Different Topical Domains. In *Proc. ISWC*, 2014.

RDF Introduction

- Everything is an uniquely named resource
- Namespaces can be used to scope the names
- Properties of resources can be defined
- Relationships with other resources can be defined
- Resources can be contributed by different people/groups and can be located anywhere in the web
  - Integrated web “database”

xmlns:y=http://en.wikipedia.org/wiki

y:Abraham_Lincoln



Abraham_Lincoln:hasName “Abraham Lincoln”
Abraham_Lincoln:BornOnDate: “1809-02-12”
Abraham_Lincoln:DiedOnDate: “1865-04-15”

Abraham_Lincoln:DiedIn

y:Washington_DC

RDF Data Model

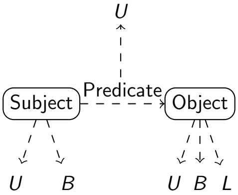
- Triple: Subject, Predicate (Property), Object (s, p, o)

  Subject: the entity that is described (URI or blank node)

  Predicate: a feature of the entity (URI)

  Object: value of the feature (URI, blank node or literal)

- $(s, p, o) \in (U \cup B) \times U \times (U \cup B \cup L)$
- Set of RDF triples is called an RDF graph



U: set of URIs
B: set of blank nodes
L: set of literals

Subject	Predicate	Object
Abraham_Lincoln	hasName	"Abraham Lincoln"
Abraham_Lincoln	BornOnDate	"1809-02-12"
Abraham_Lincoln	DiedOnDate	"1865-04-15"

RDF Example Instance

Prefix: y=http://en.wikipedia.org/wiki

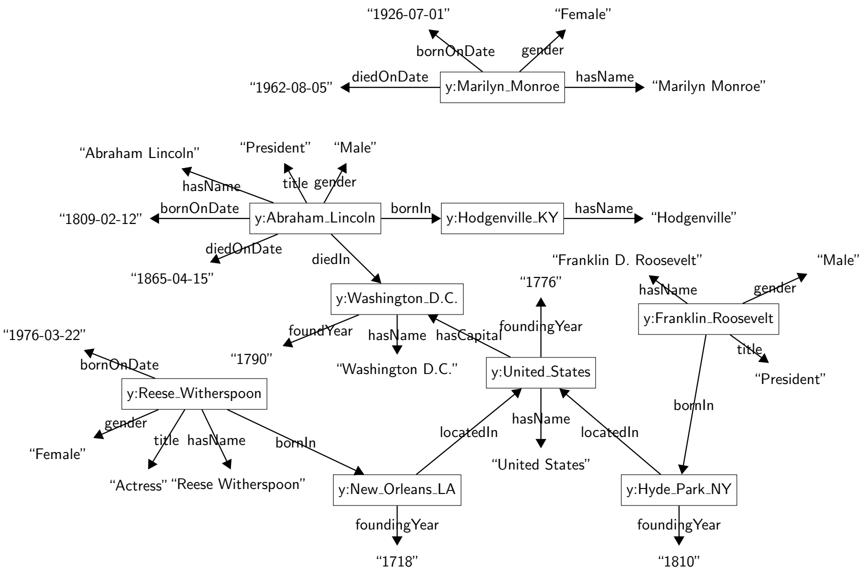
Subject	Predicate	Object
y: Abraham_Lincoln	hasName	“Abraham Lincoln”
y: Abraham_Lincoln	BornOnDate	“1809-02-12”’
y: Abraham_Lincoln	DiedOnDate	“1865-04-15”
y:Abraham_Lincoln	bornIn	y:Hodgenville_KY
y: Abraham_Lincoln	DiedIn	y: Washington_DC
y:Abraham_Lincoln	title	“President”
y:Abraham_Lincoln	gender	“Male”
y: Washington_DC	hasName	“Washington D.C.”
y:Washington_DC	foundingYear	“1790”
y:Hodgenville_KY	hasName	“Hodgenville”
y:United_States	hasName	“United States”
y:United_States	hasCapital	y:Washington_DC
y:United_States	foundingYear	“1776”
y:Reese_Witherspoon	bornOnDate	“1976-03-22”
y:Reese_Witherspoon	bornIn	y:New_Orleans_LA
y:Reese_Witherspoon	hasName	“Reese Witherspoon”
y:Reese_Witherspoon	gender	“Female”
y:Reese_Witherspoon	title	“Actress”
y:New_Orleans_LA	foundingYear	“1718”
y:New_Orleans_LA	locatedIn	y:United_States
y:Franklin_Roosevelt	hasName	“Franklin D. Roosevelt”
y:Franklin_Roosevelt	bornIn	y:Hyde_Park_NY
y:Franklin_Roosevelt	title	“President”
y:Franklin_Roosevelt	gender	“Male”
y:Hyde_Park_NY	foundingYear	“1810”
y:Hyde_Park_NY	locatedIn	y:United_States
y:Marilyn_Monroe	gender	“Female”
y:Marilyn_Monroe	hasName	“Marilyn Monroe”
y:Marilyn_Monroe	bornOnDate	“1926-07-01”
y:Marilyn_Monroe	diedOnDate	“1962-08-05”

Literal

URI

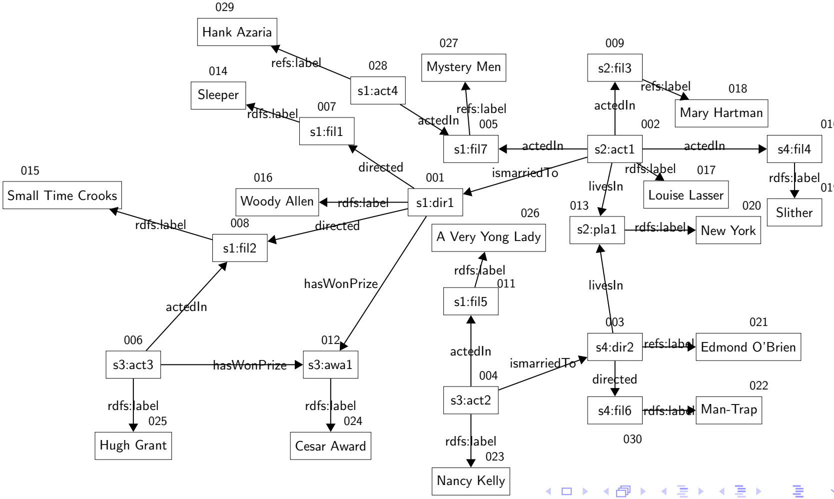
URI

RDF Graph



"1926-07-01"
"Female"
bornOnDate
gender
diedOnDate
y:Marilyn_Monroe
hasName
"1962-08-05"
"Marilyn Monroe"
"Abraham Lincoln"
"President"
"Male"
hasName
title
gender
bornOnDate
y:Abraham_Lincoln
bornIn
y:Hodgenville_KY
hasName
"Hodgenville"
"1809-02-12"
diedOnDate
diedIn
"1865-04-15"
"Franklin D. Roosevelt"
"Male"
"1776"
hasName
gender
y:Washington_D.C.
y:Franklin_Roosevelt
foundYear
hasName
hasCapital
foundingYear
"1976-03-22"
title
bornOnDate
"1790"
"Washington D.C."
y:United_States
"President"
y:Reese_Witherspoon
bornIn
gender
hasName
locatedIn
locatedIn
title
hasName
bornIn
"Female"
"United States"
"Actress"
"Reese Witherspoon"
y:New_Orleans_LA
y:Hyde_Park_NY
foundingYear
foundingYear
"1718"
"1810"

A Distributed RDF Graph



Representative graph processing systems

	Property graphs	Online query	Data sharding	In-memory storage	Atomicity & Transaction
★ Neo4j	Yes	Yes	No	No	Yes
★ Trinity	Yes	Yes	Yes	Yes	Atomicity
★ Horton	Yes	Yes	Yes	Yes	No
★ HyperGraphDB	No	Yes	No	No	Yes
★ FlockDB	No	Yes	Yes	No	Yes
★ TinkerGraph	Yes	Yes	No	Yes	No
★ InfiniteGraph	Yes	Yes	Yes	No	Yes
★ Cayley	Yes	Yes	SB	SB	Yes
★ Titan	Yes	Yes	SB	SB	Yes
★ MapReduce	No	No	Yes	No	No
★ PEGASUS	No	No	Yes	No	No
★ Pregel	No	No	Yes	No	No
★ Giraph	No	No	Yes	No	No
★ GraphLab	No	No	Yes	No	No
★ GraphChi	No	No	No	No	No
★ GraphX	No	No	Yes	No	No

DB-Engines Ranking of Graph DBMS

- Cypher query language is used by Neo4j.
- Gremlin is used by most of graph DBMSs.
- GSQL is used by TigerGraph.

31 systems in ranking, September 2018

Rank Sep 2018	Rank Aug 2018	Rank Sep 2017	DBMS	Database Model	Score Sep 2018	Score Aug 2018	Score Sep 2017
1.	1.	1.	Neo4j	Graph DBMS	40.10	-0.83	+1.67
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model	19.18	-0.35	+7.95
3.	3.		Datastax Enterprise	Multi-model	7.76	+0.46	
4.	4.	↓3.	OrientDB	Multi-model	5.48	+0.57	-0.42
5.	5.	5.	ArangoDB	Multi-model	4.05	+0.71	+1.05
6.	6.	6.	Virtuoso	Multi-model	2.06	+0.01	+0.17
7.	↑8.		Amazon Neptune	Multi-model	1.12	+0.31	
8.	↓7.	↓7.	Giraph	Graph DBMS	1.02	+0.03	-0.05
9.	↑11.	↑16.	JanusGraph	Graph DBMS	0.90	+0.36	+0.68
10.	10.	↓9.	GraphDB	Multi-model	0.63	+0.06	+0.02
11.	↓9.	↓8.	AllegroGraph	Multi-model	0.60	+0.02	-0.04
12.	12.	↓10.	Stardog	Multi-model	0.54	+0.01	-0.04
13.	↑17.	13.	Dgraph	Graph DBMS	0.41	+0.17	+0.14
14.	↑15.	↑15.	Blazegraph	Multi-model	0.36	+0.08	+0.12
15.	↓13.	↓11.	Sqrrl	Multi-model	0.34	-0.00	-0.17
16.	16.	↓14.	Graph Engine	Multi-model	0.29	+0.02	+0.02
17.	↓14.	↓12.	InfiniteGraph	Graph DBMS	0.28	-0.02	-0.01
18.	18.		TigerGraph	Graph DBMS	0.22	+0.02	
19.	19.	19.	FaunaDB	Multi-model	0.17	+0.02	+0.00
20.	↑21.	↑22.	VelocityDB	Multi-model	0.14	+0.01	+0.03

https://db-engines.com/en/ranking/graph+dbms

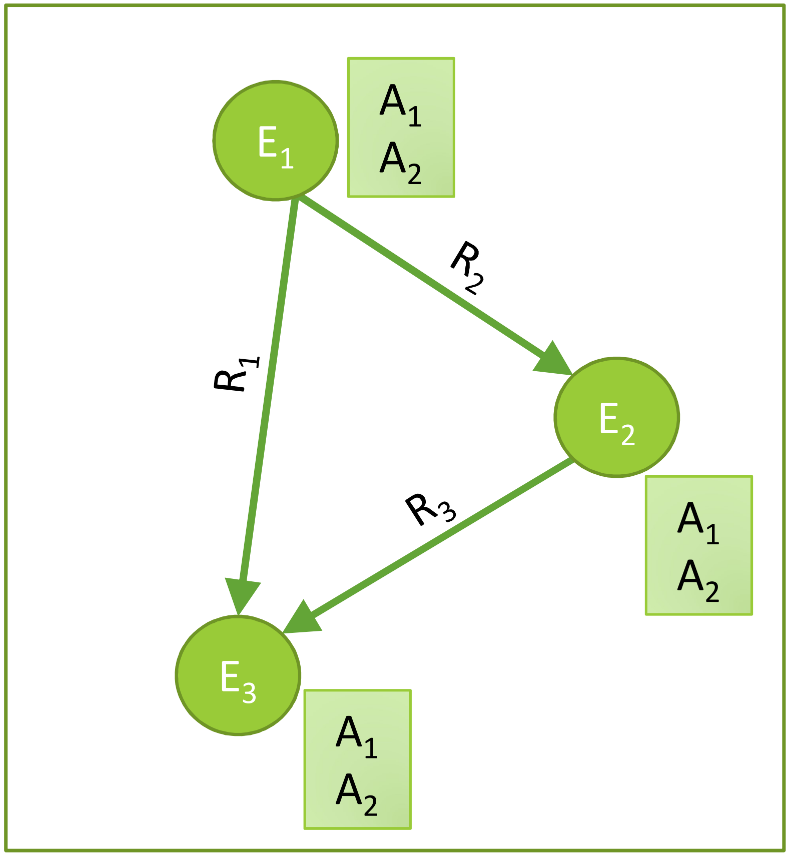
Knowledge Graph Primer

TOPICS:

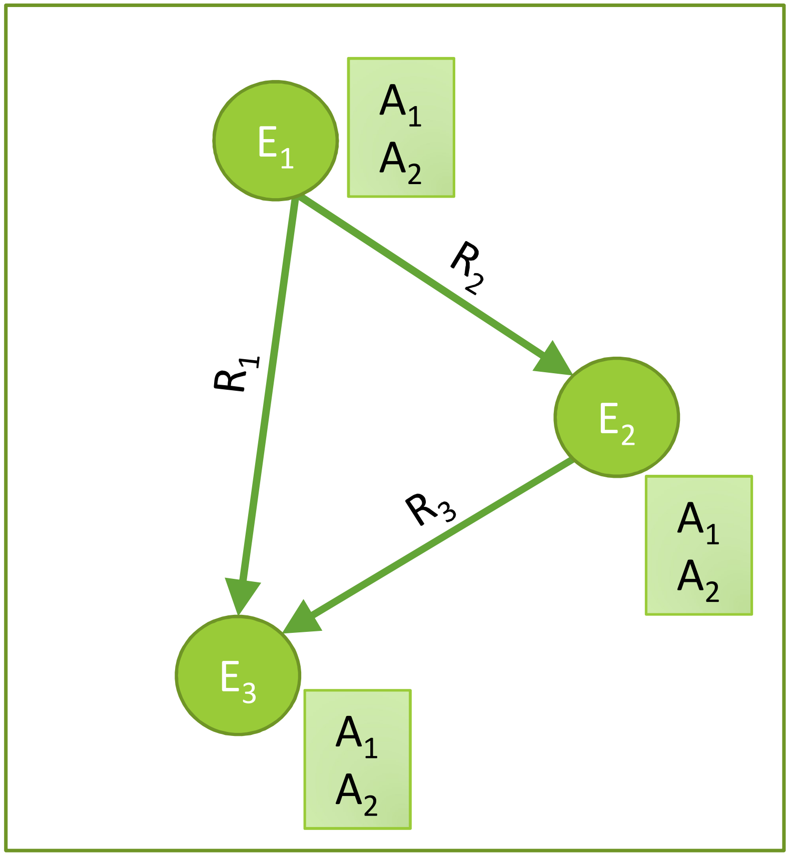
- What is a Knowledge Graph?
- Why are Knowledge Graphs Important?
- Where do Knowledge Graphs come from?
- Knowledge Representation Choices
- Problem Overview

What is a knowledge graph?

- Knowledge in graph form!
- Captures entities, attributes, and relationships
- Nodes are entities
- Nodes are labeled with attributes (e.g., types)
- Typed edges between two nodes capture a relationship between entities

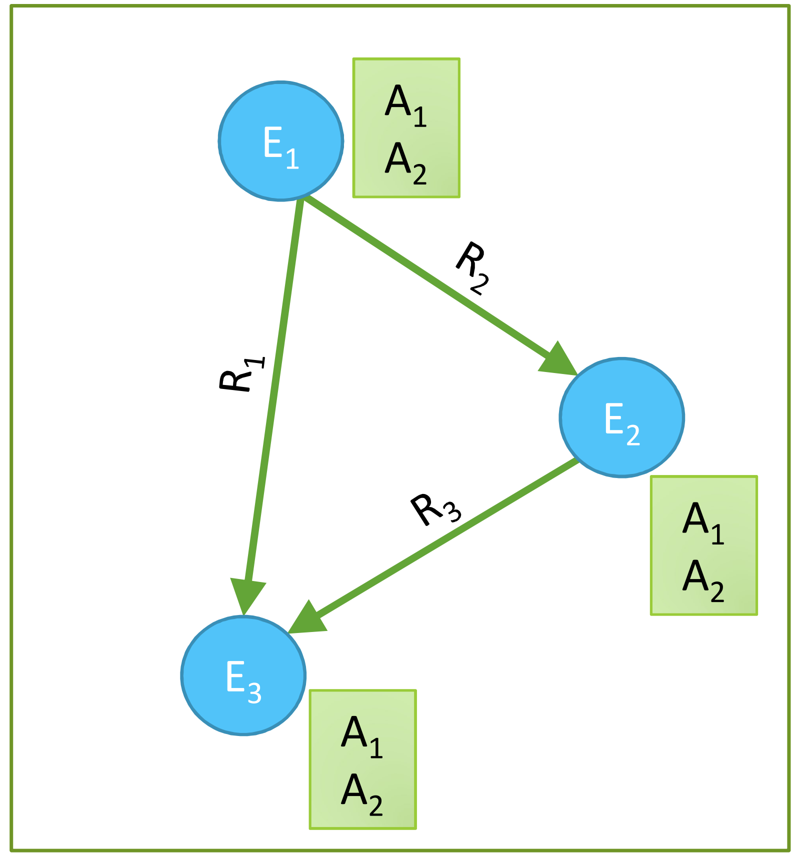


Basic problems



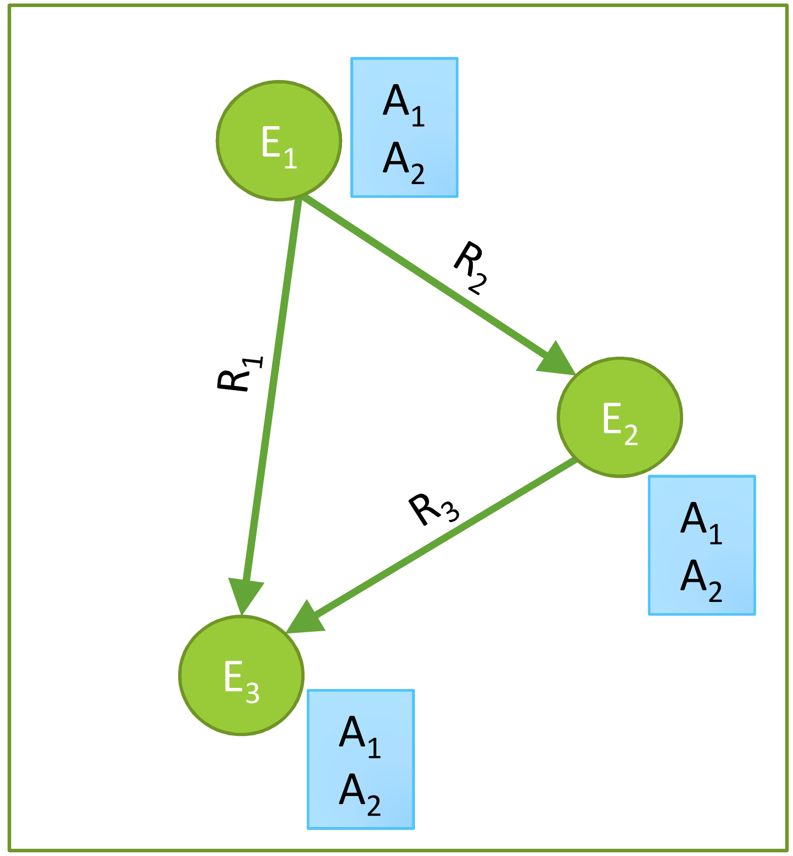
Basic problems

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



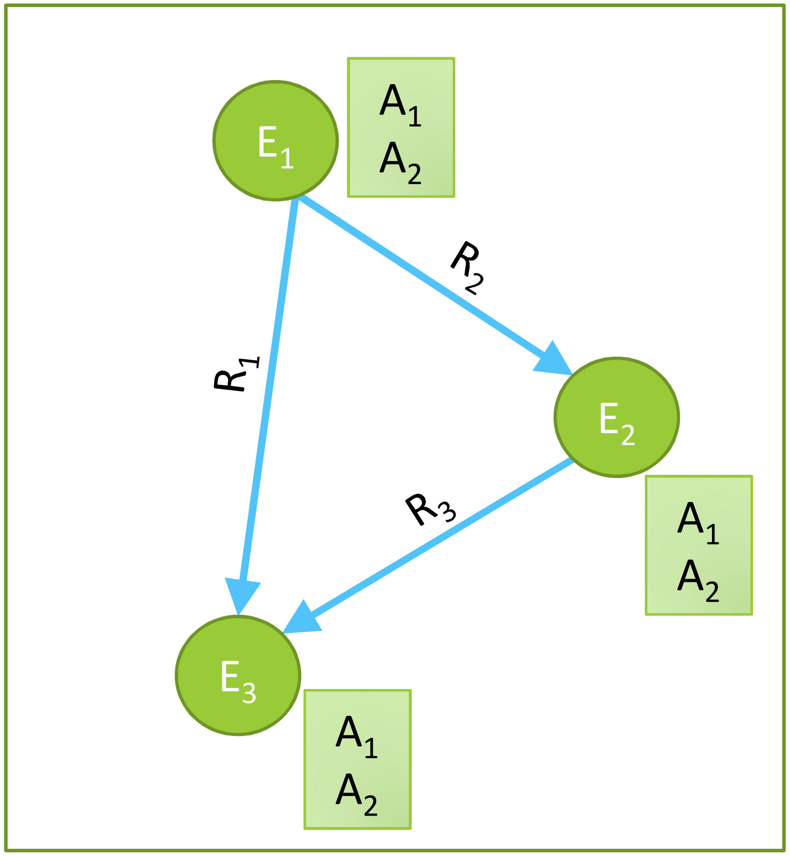
Basic problems

- **Who** are the entities (nodes) in the graph?
- **What** are their attributes and types (labels)?



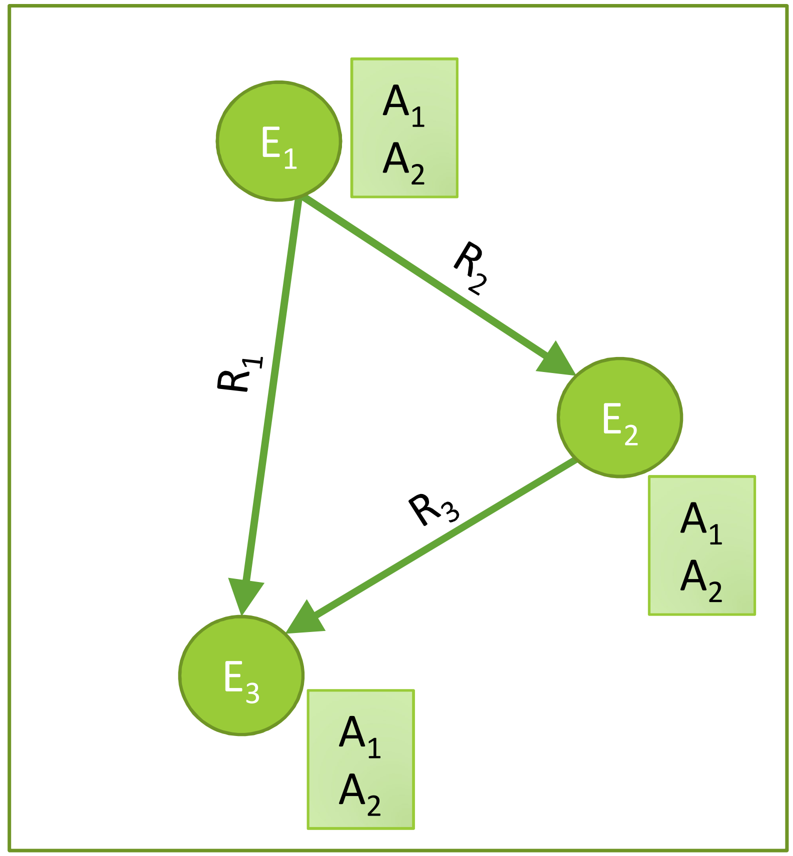
Basic problems

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

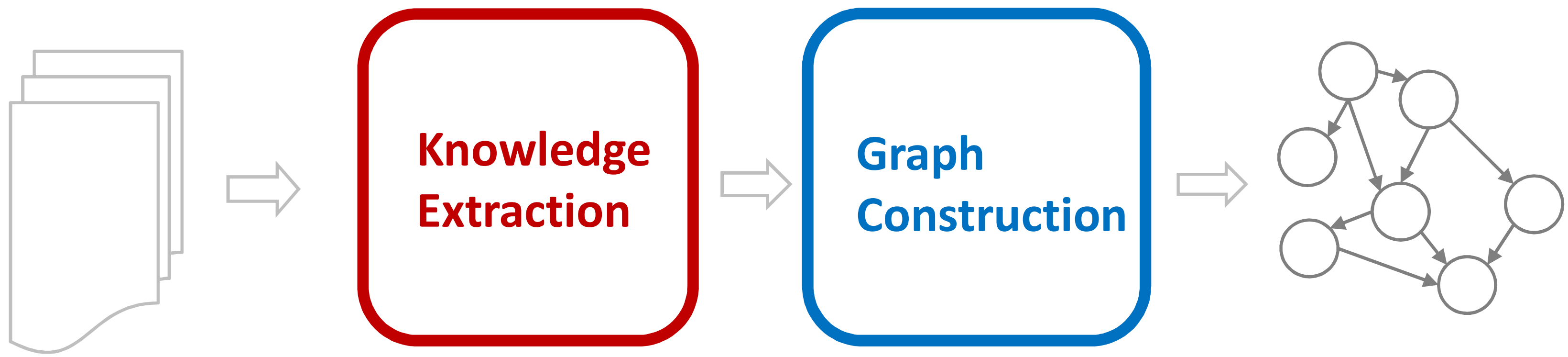


Basic problems

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



Knowledge Graph Construction



Two perspectives

	Extraction graph	Knowledge graph
Who are the entities? (nodes)	• Named Entity Recognition • Entity Coreference	• Entity Linking • Entity Resolution
What are their attributes? (labels)	• Entity Typing	• Collective classification
How are they related? (edges)	• Semantic role labeling • Relation Extraction	• Link prediction

What is NLP?

Unstructured
Ambiguous
Lots and lots of it!

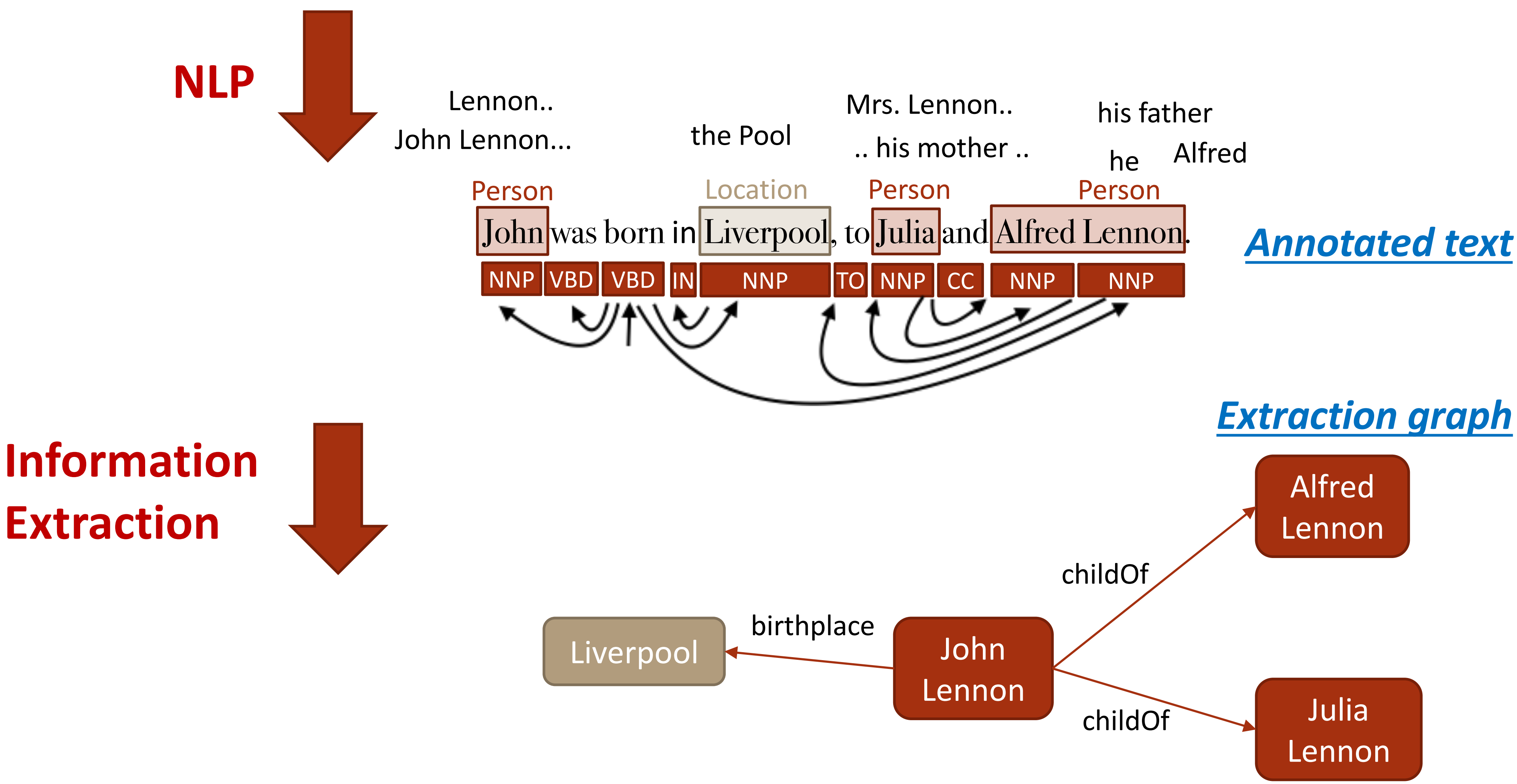
Structured
Precise, Actionable
Specific to the task

Humans can read them, but
… very slowly
… can’t remember all
… can’t answer questions

Can be used for downstream applications, such as creating Knowledge Graphs!

Knowledge Extraction

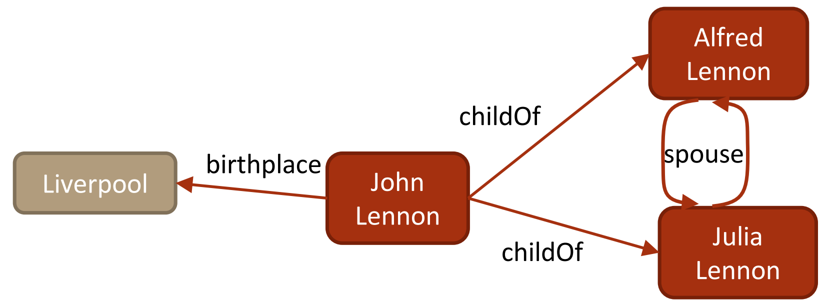
John was born in Liverpool, to Julia and Alfred Lennon. *Text*



Breaking it Down

Information Extraction

Entity resolution,
Entity linking,
Relation extraction…



Document

Coreference Resolution…

Lennon..
John Lennon…

the Pool

Mrs. Lennon..
.. his mother ..

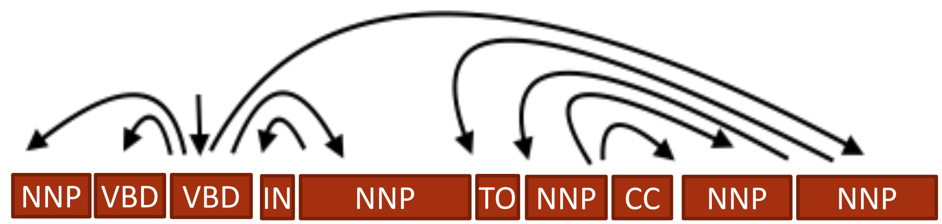
his father
he
Alfred

Person: John | Location: Liverpool | Person: Julia | Person: Alfred Lennon

John was born in Liverpool, to Julia and Alfred Lennon.

Sentence

Dependency Parsing,
Part of speech tagging,
Named entity recognition…



John was born in Liverpool, to Julia and Alfred Lennon.

Tagging the Parts of Speech

NNP	VBD	VBD	IN	NNP	TO	NNP	CC	NNP	NNP
John	was	born	in	Liverpool,	to	Julia	and	Alfred	Lennon.

Nouns are entities

Verbs are relations

- Common approaches include CRFs, CNNs, LSTMs

Detecting Named Entities

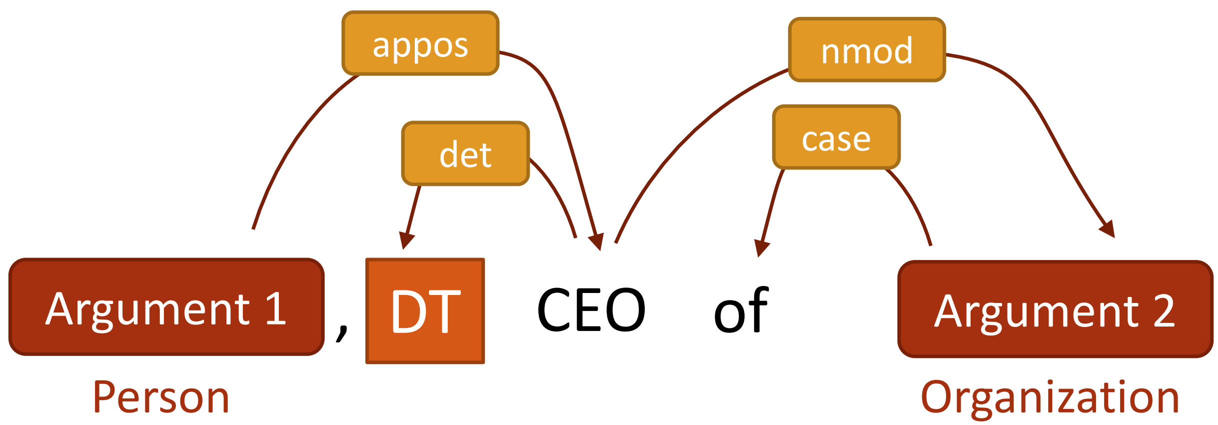
Person Location Person Person

John was born in Liverpool, to Julia and Alfred Lennon.

- Structured prediction approaches
- Capture entity mentions and entity types

NLP annotations → features for IE

Combine tokens, dependency paths, and entity types to define rules.



Bill Gates, the CEO of Microsoft, said …
Mr. Jobs, the brilliant and charming CEO of Apple Inc., said …
… announced by Steve Jobs, the CEO of Apple.
… announced by Bill Gates, the director and CEO of Microsoft.
… mused Bill, a former CEO of Microsoft.
and many other possible instantiations…

Entity Names: Two Main Problems

Entities with Same Name

Same type of entities share names

Kevin Smith, John Smith, Springfield, …

Things named after each other

Clinton, Washington, Paris, Amazon, Princeton, Kingston, …

Partial Reference

First names of people, Location instead of team name, Nick names

Different Names for Entities

Nick Names

Bam Bam, Drumpf, …

Typos/Misspellings

Baarak, Barak, Barrack, …

Inconsistent References

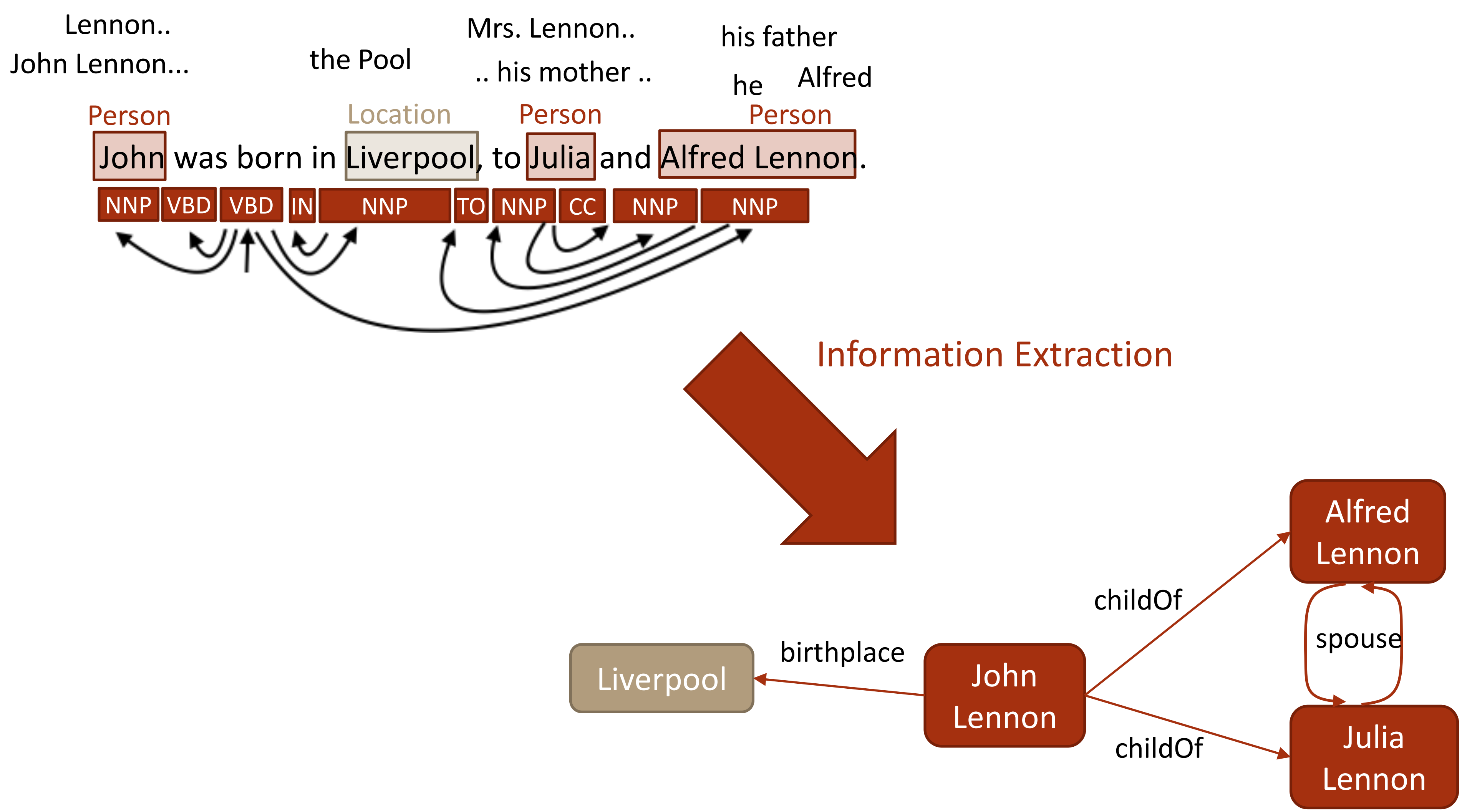
MSFT, APPL, GOOG…

Entity Linking Approach

Washington drops 10 points after game with UCLA Bruins.

Candidate Generation		Washington DC, George Washington, Washington state, Lake Washington, Washington Huskies, Denzel Washington, University of Washington, Washington High School, …
Entity Types	LOC/ORG	Washington DC, ~~George Washington~~, Washington state, Lake Washington, Washington Huskies, ~~Denzel Washington~~, University of Washington, Washington High School, …
Coreference	UWashington, Huskies	~~Washington DC~~, ~~George Washington~~, ~~Washington state~~, ~~Lake Washington~~, Washington Huskies, ~~Denzel Washington~~, University of Washington, ~~Washington High School~~, …
Coherence	UCLA Bruins, USC Trojans	~~Washington DC~~, ~~George Washington~~, ~~Washington state~~, ~~Lake Washington~~, Washington Huskies, ~~Denzel Washington~~, ~~University of Washington~~, ~~Washington High School~~, …

Information Extraction



Information Extraction

3 CONCRETE SUB-PROBLEMS

Defining domain

Learning extractors

Scoring the facts

×

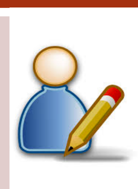
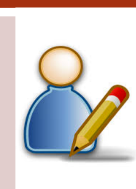
3 LEVELS OF SUPERVISION

Supervised 

Semi-supervised 

Unsupervised 

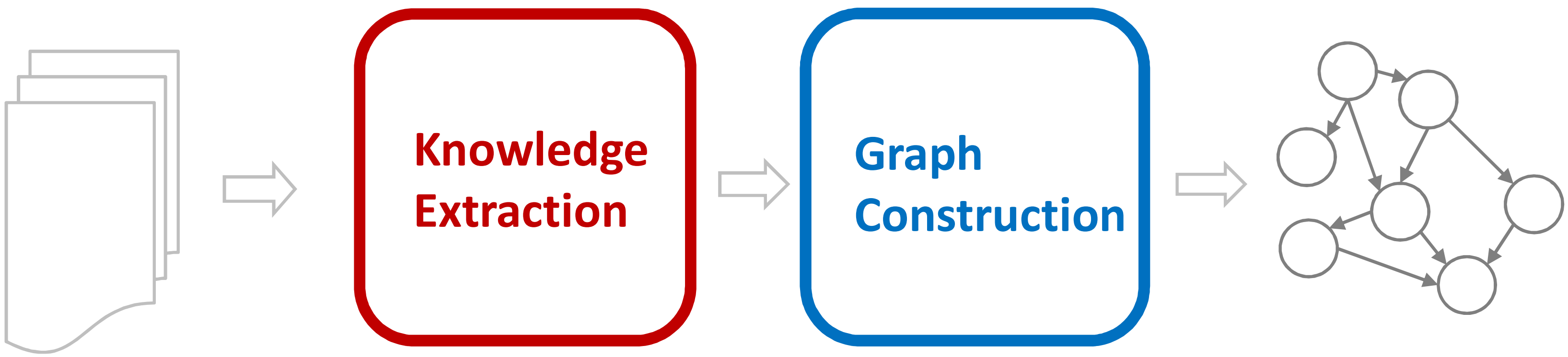
IE systems in practice

	Defining domain	Learning extractors	Scoring candidate facts	Fusing extractors
ConceptNet				
NELL				Heuristic rules
Knowledge Vault				Classifier
OpenIE				

Knowledge Extraction: Key Points

- Built on the foundation of NLP techniques
  - Part-of-speech tagging, dependency parsing, named entity recognition, coreference resolution…
  - Challenging problems with very useful outputs
- Information extraction techniques use NLP to:
  - define the domain
  - extract entities and relations
  - score candidate outputs
- Trade-off between manual & automatic methods

Knowledge Graph Construction



Knowledge Graph Construction

TOPICS:

PROBLEM SETTING

PROBABILISTIC MODELS

EMBEDDING TECHNIQUES

Knowledge Graph Construction

TOPICS:

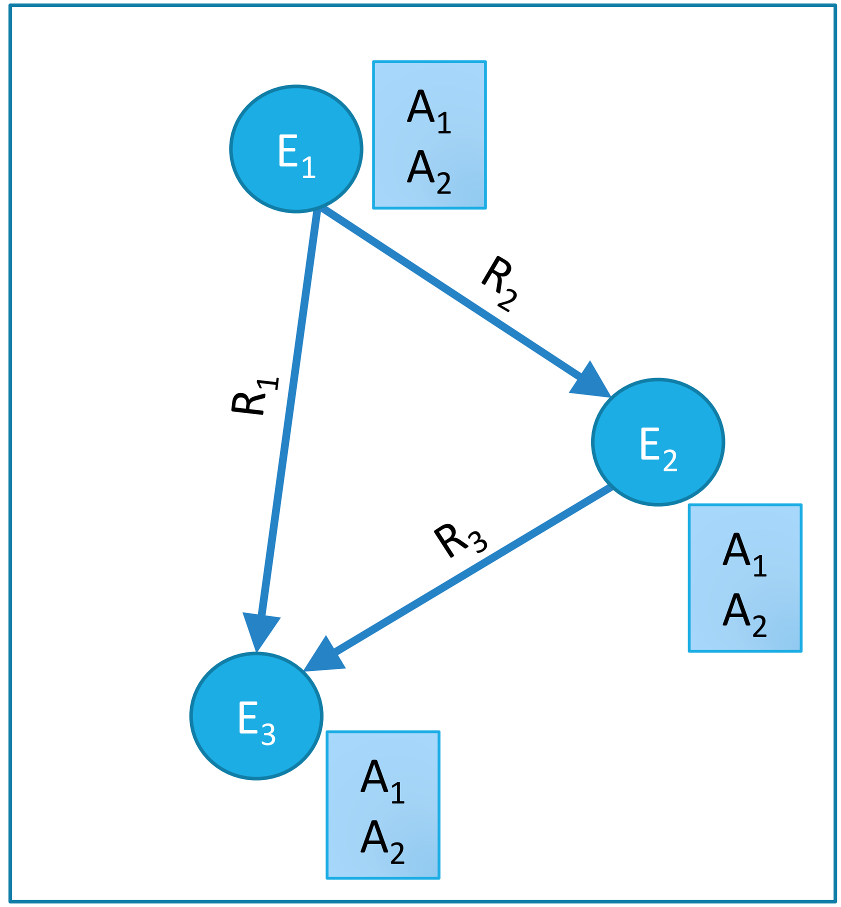
<u>PROBLEM SETTING</u>

PROBABILISTIC MODELS

EMBEDDING TECHNIQUES

Reminder: Basic problems

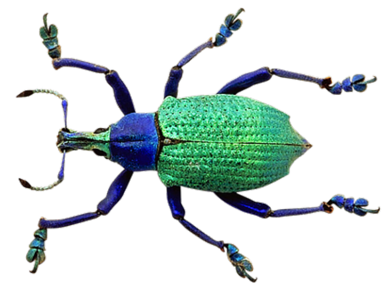
- **Who** are the entities (nodes) in the graph?
- **What** are their attributes and types (labels)?
- **How** are they related (edges)?



Graph Construction Issues

Extracted knowledge is:

- ambiguous:
  - Ex: Beetles, beetles, Beatles
  - Ex: citizenOf, livedIn, bornIn

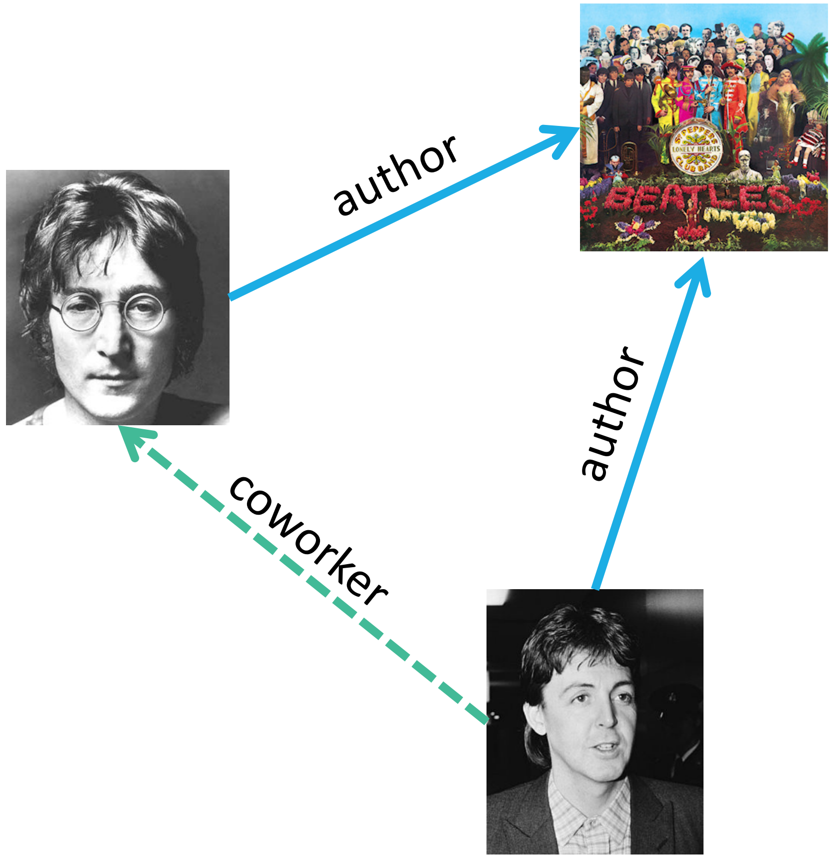


Graph Construction Issues

Extracted knowledge is:

- ambiguous

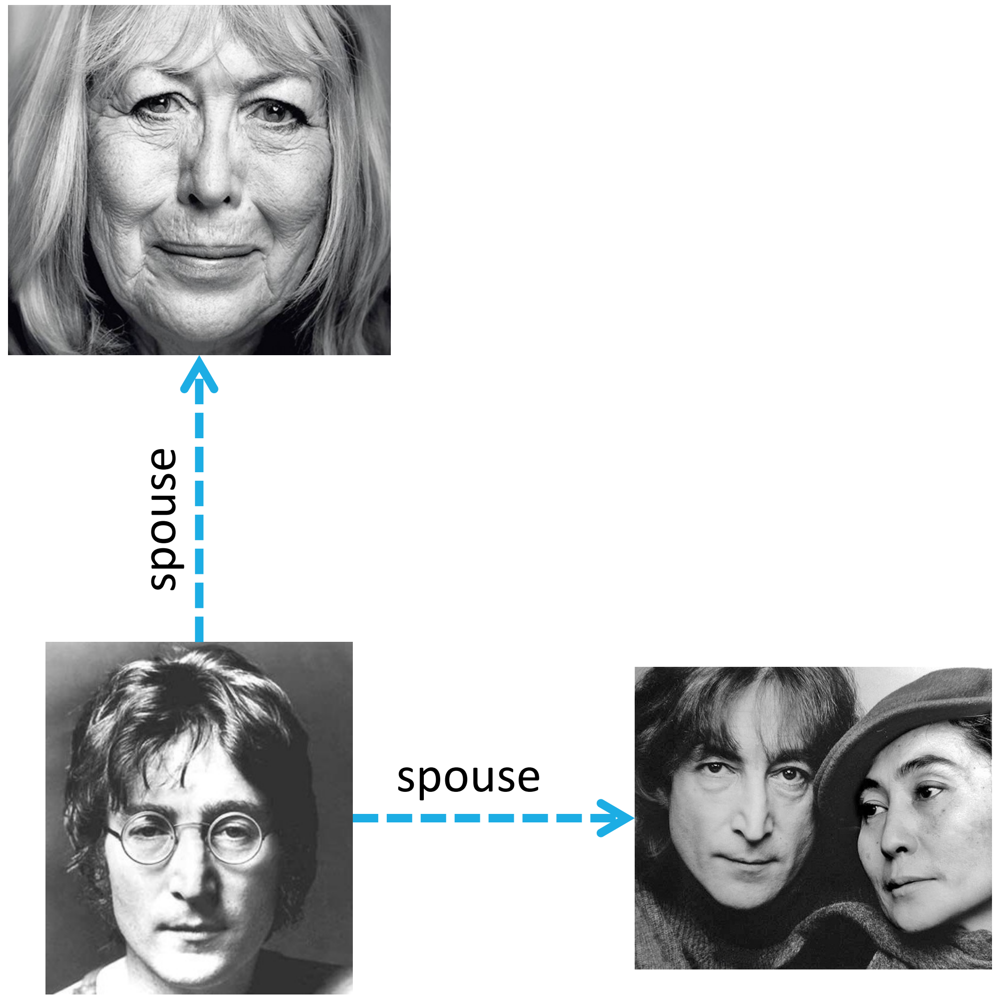
- incomplete
  - Ex: missing relationships
  - Ex: missing labels
  - Ex: missing entities



Graph Construction Issues

Extracted knowledge is:

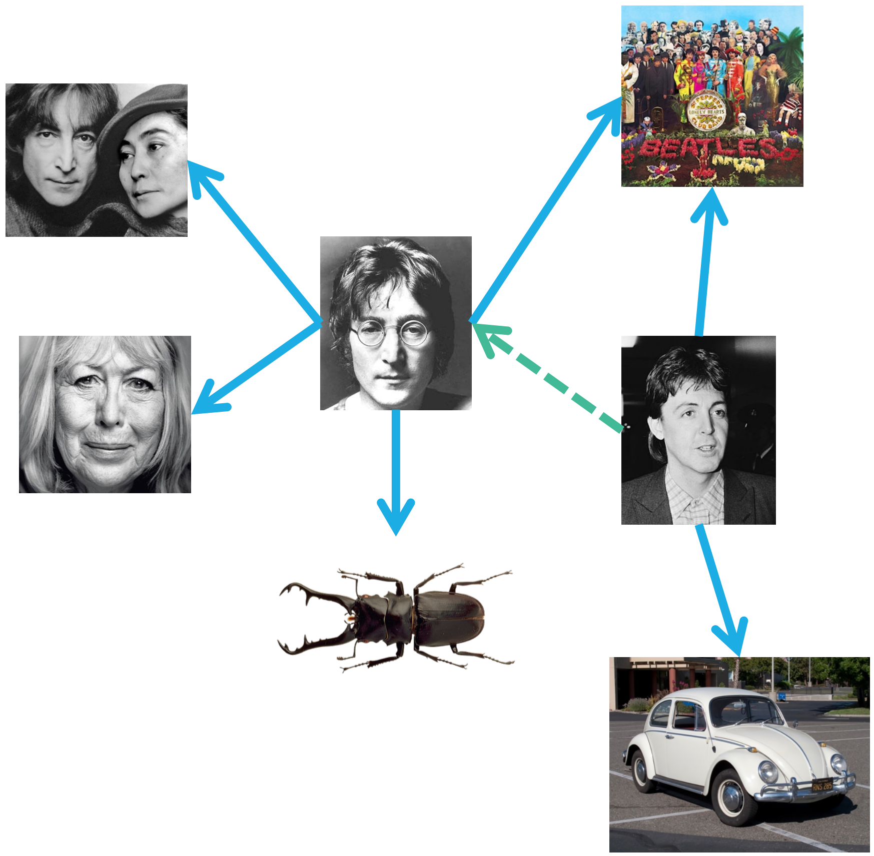
- ambiguous
- incomplete
- inconsistent
  - Ex: Cynthia Lennon, Yoko Ono
  - Ex: exclusive labels (alive, dead)
  - Ex: domain-range constraints



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

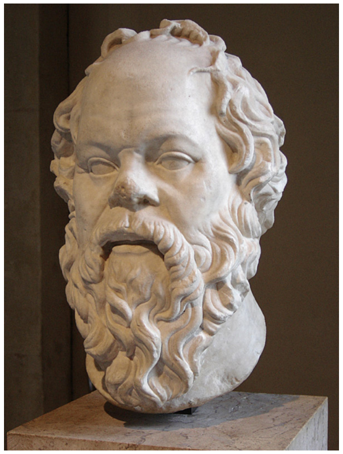
TOPICS:

OVERVIEW

GRAPHICAL MODELS

RANDOM WALK METHODS

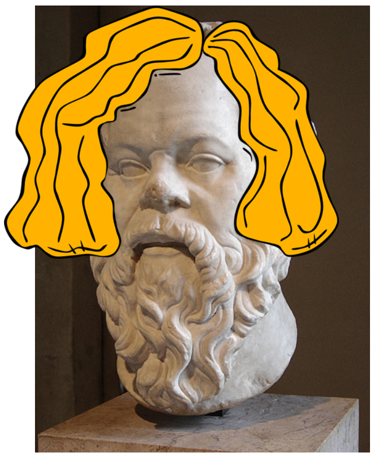
Beyond Pure Reasoning



- Classical AI approach to knowledge: reasoning

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

Beyond Pure Reasoning

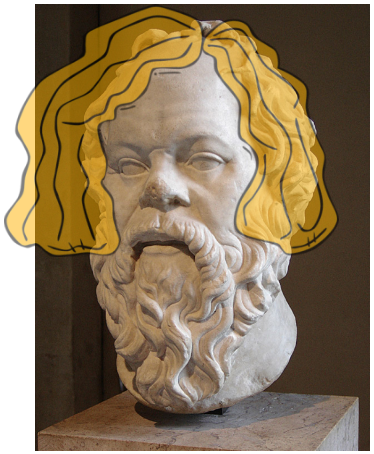


- Classical AI approach to knowledge: reasoning

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

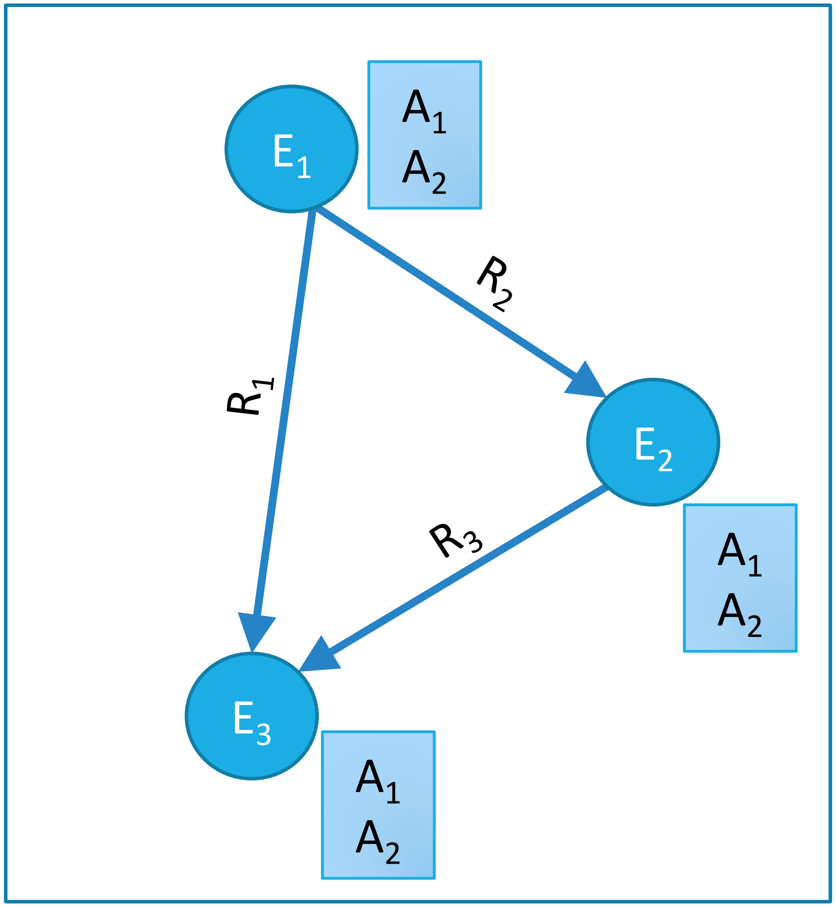
Graphical Models: Overview

- 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

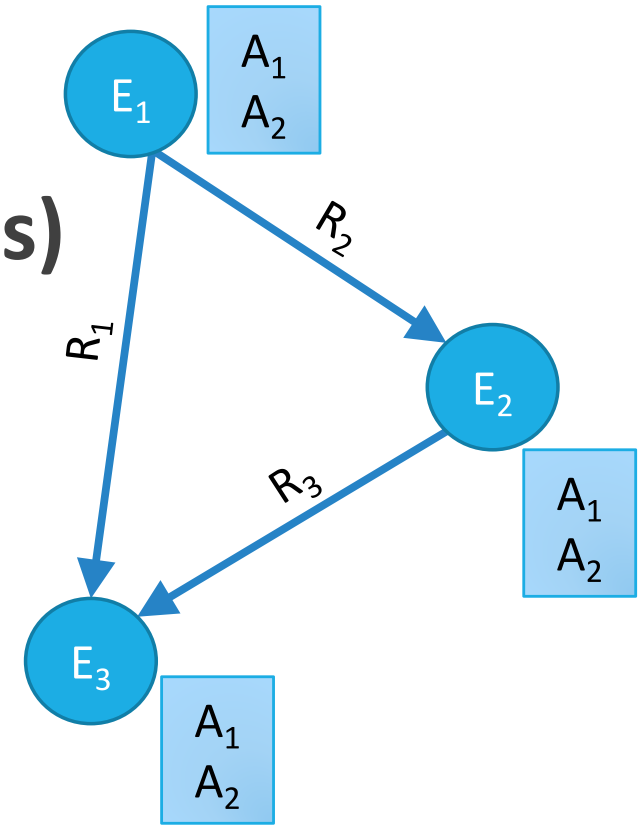
Define a graphical model to perform all three of these tasks simultaneously!

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



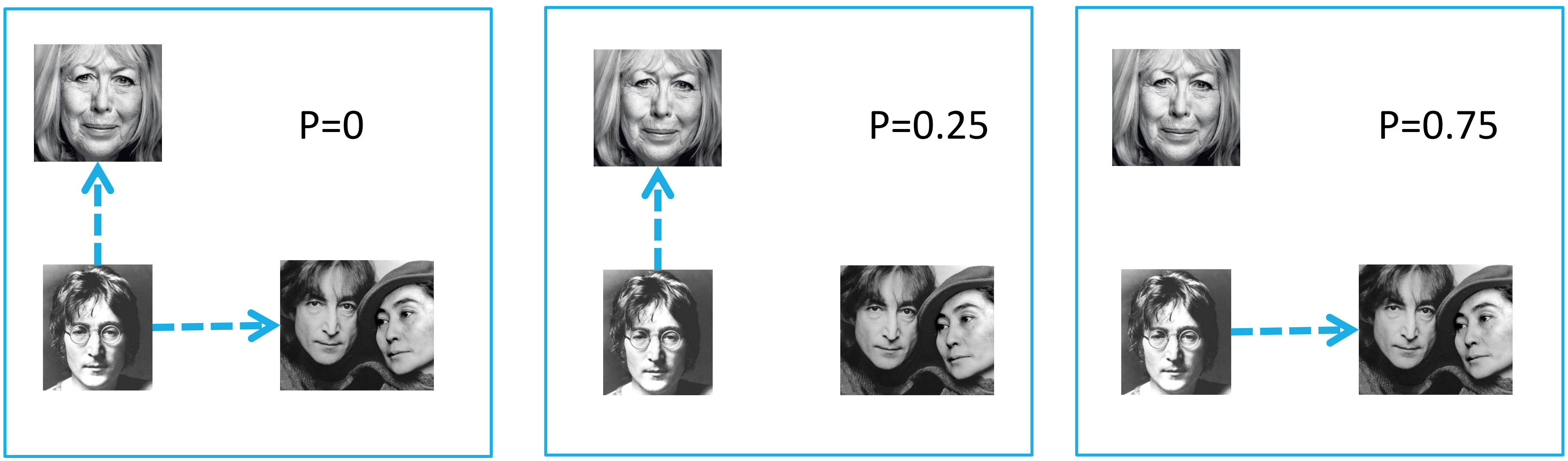
Knowledge Graph Identification

P(Who, What, How | 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(R(John,Spouse,Yoko))=0.75; P(R(John,Spouse,Cynthia))=0.25
  - 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,Spouse,Yoko))=0.75; P(R(John,Spouse,Cynthia))=0.25
  - LevenshteinSimilarity(Beatles, Beetles) = 0.9

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

- **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)

Example: The Fab Four

THE 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)

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)

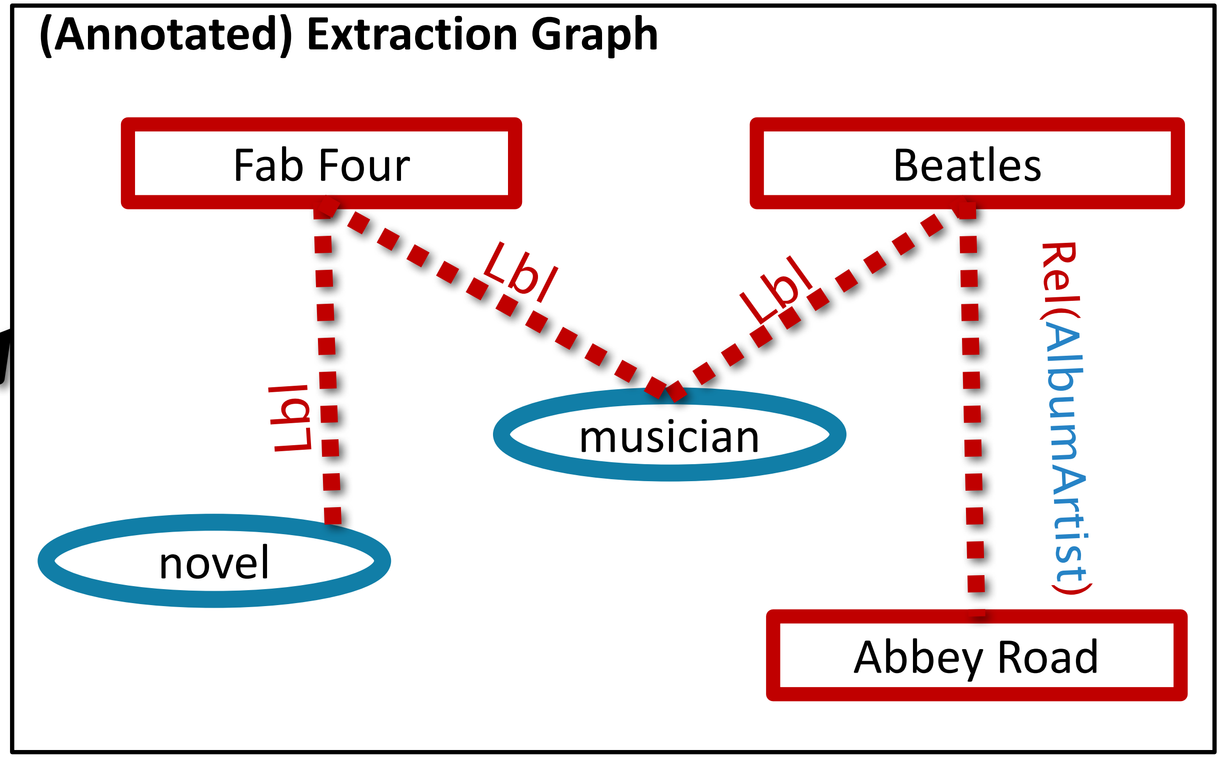


Illustration of KG Identification

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)

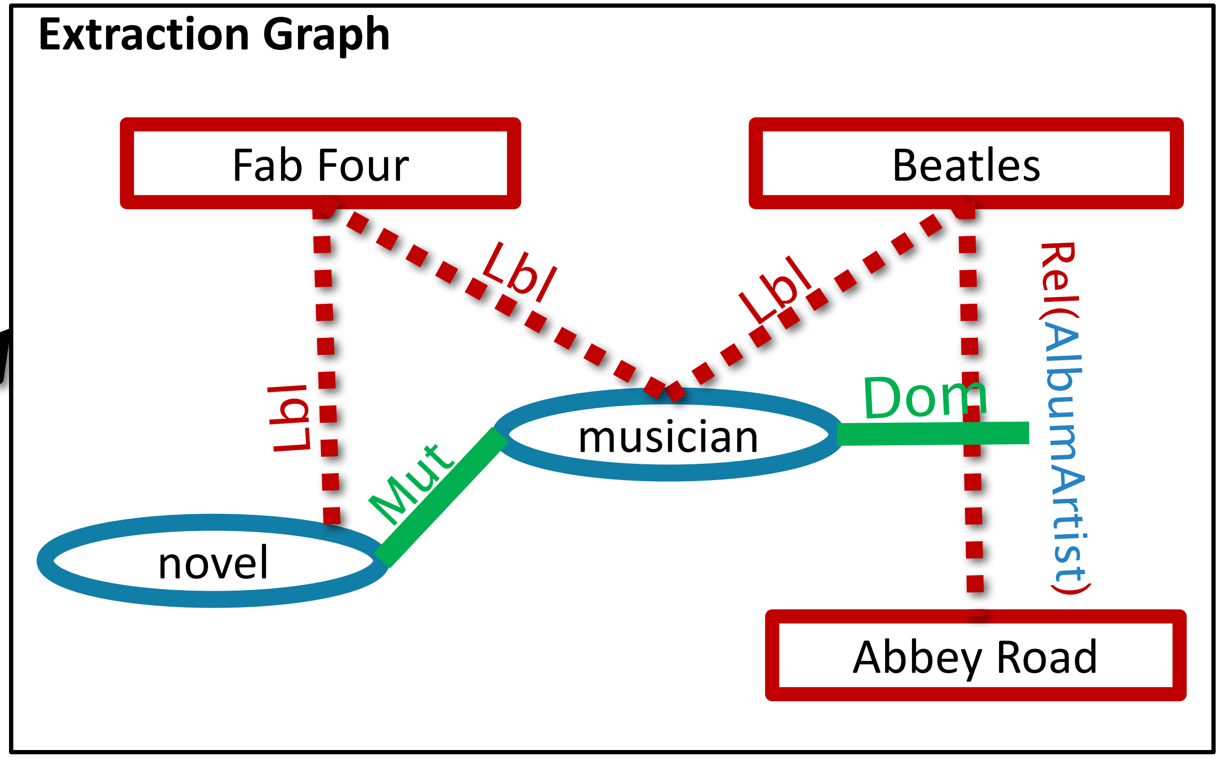


Illustration of KG Identification

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)

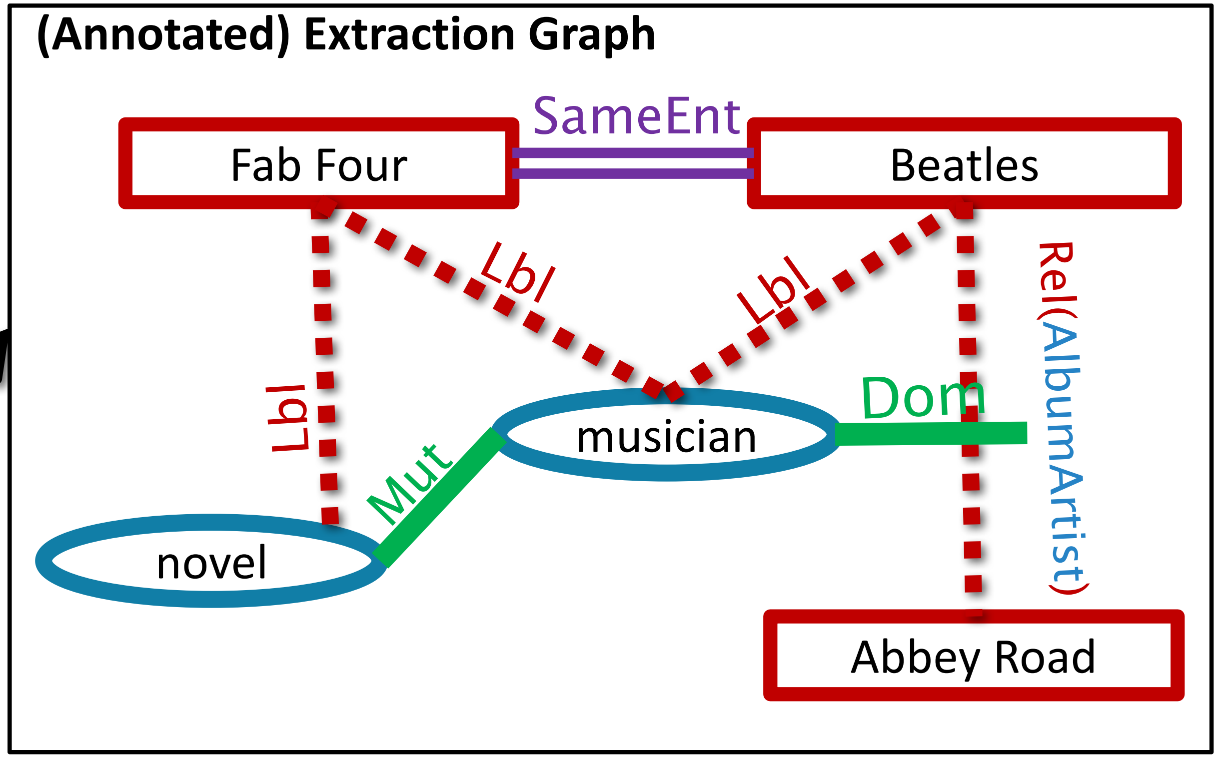
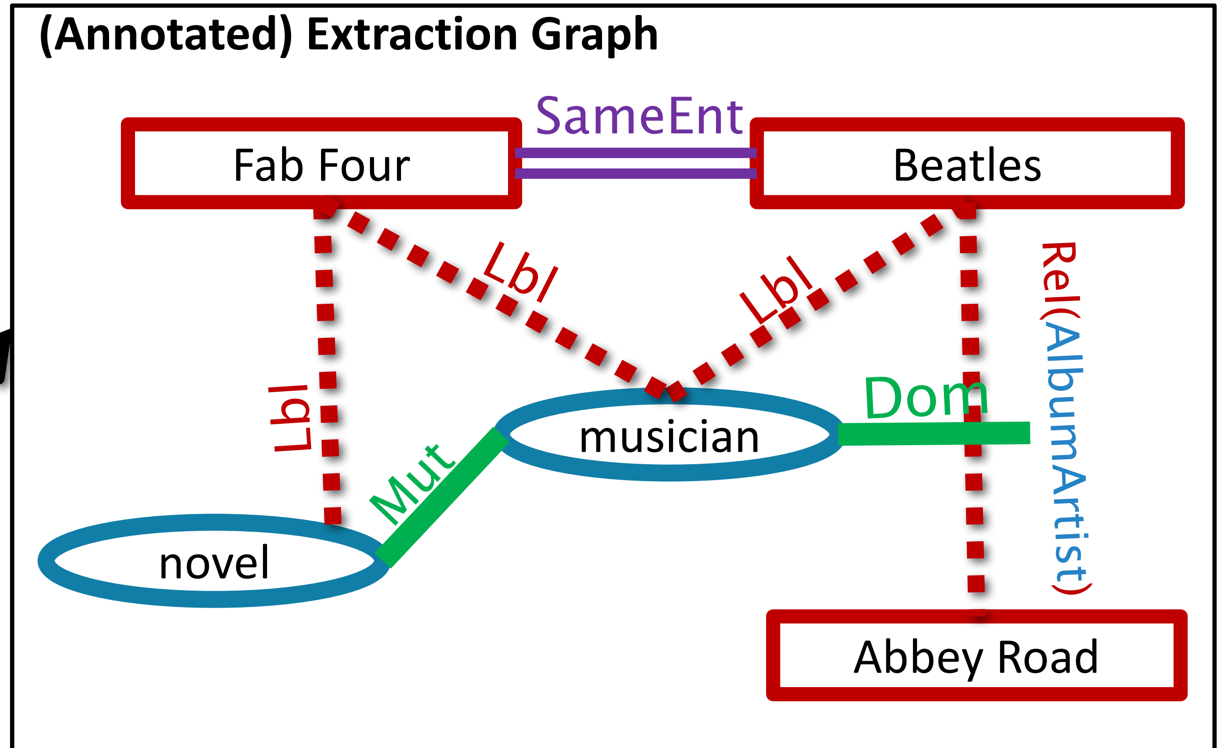


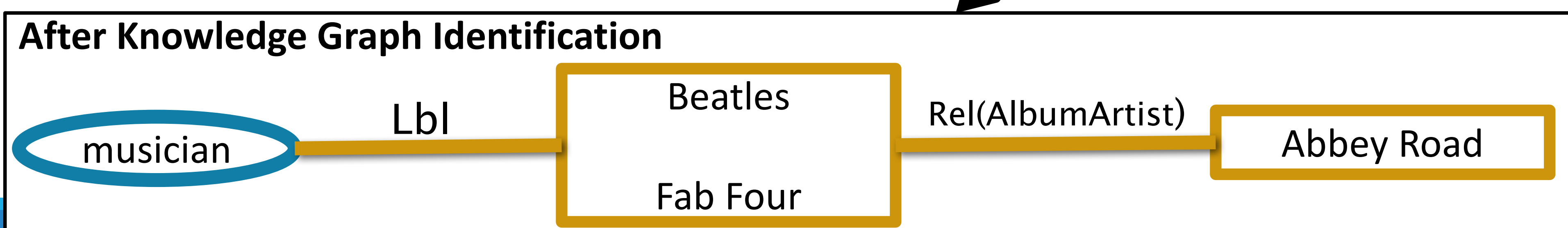
Illustration of KG Identification

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)

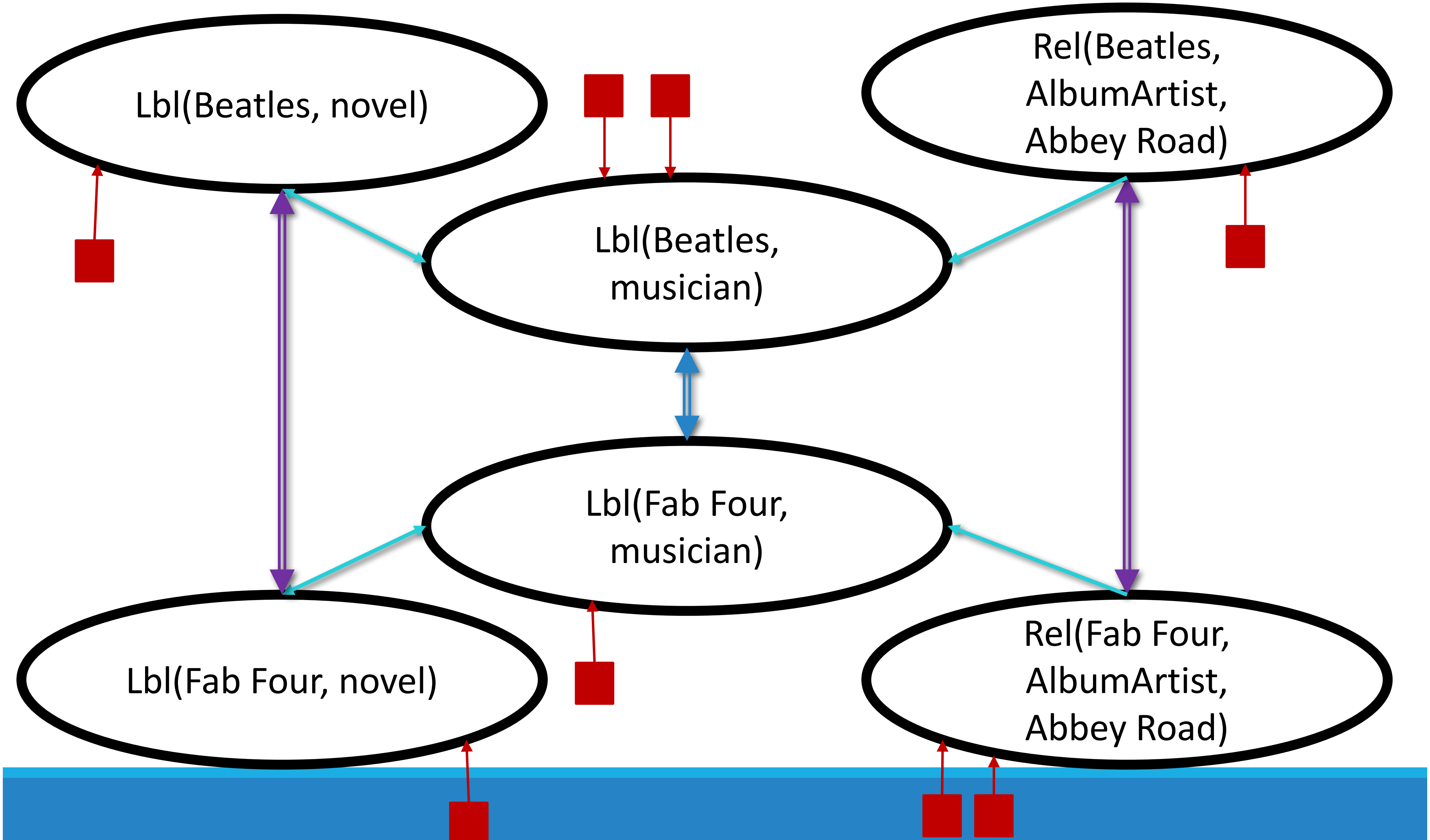
Entity Resolution:
SameEnt(Fab Four, Beatles)





PUJARA+ISWC13; PUJARA+AIMAG15

Probabilistic graphical model for KG



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

```
100:    Subsumes(L1,L2)   & Label(E,L1)         ->  Label(E,L2)
100:    Exclusive(L1,L2)  & Label(E,L1)         -> !Label(E,L2)

100:    Inverse(R1,R2)    & Relation(R1,E,O)    ->  Relation(R2,O,E)
100:    Subsumes(R1,R2)   & Relation(R1,E,O)    ->  Relation(R2,E,O)
100:    Exclusive(R1,R2)  & Relation(R1,E,O)    -> !Relation(R2,E,O)

100:    Domain(R,L)       & Relation(R,E,O)     ->  Label(E,L)
100:    Range(R,L)        & Relation(R,E,O)     ->  Label(O,L)

10:     SameEntity(E1,E2) & Label(E1,L)         ->  Label(E2,L)
10:     SameEntity(E1,E2) & Relation(R,E1,O)    ->  Relation(R,E2,O)

1:      Label_OBIE(E,L)                         ->  Label(E,L)
1:      Label_OpenIE(E,L)                       ->  Label(E,L)
1:      Relation_Pattern(R,E,O)                 ->  Relation(R,E,O)
1:                                                 !Relation(R,E,O)
1:                                                 !Label(E,L)
```

Rules to Distributions

- Rules are *grounded* by substituting literals into formulas

$$\mathbf{w_r} : \textsc{SameEnt}(\text{Fab Four}, \text{Beatles}) \ \wedge$$
$$\textsc{Lbl}(\text{Beatles}, \text{musician}) \Rightarrow \textsc{Lbl}(\text{Fab Four}, \text{musician})$$

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

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

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

Probability Distribution over KGs

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

$$\left\{\begin{array}{ll}
\textsc{CandLbl}_T(\texttt{FabFour},\texttt{novel}) & \Rightarrow\ \textsc{Lbl}(\texttt{FabFour},\texttt{novel}) \\[1em]
\textsc{Mut}(\texttt{novel},\texttt{musician}) & \wedge\ \textsc{Lbl}(\texttt{Beatles},\texttt{novel}) \\
 & \Rightarrow\ \neg\textsc{Lbl}(\texttt{Beatles},\texttt{musician}) \\[1em]
\textsc{SameEnt}(\texttt{Beatles},\texttt{FabFour}) & \wedge\ \textsc{Lbl}(\texttt{Beatles},\texttt{musician}) \\
 & \Rightarrow\ \textsc{Lbl}(\texttt{FabFour},\texttt{musician})
\end{array}\right.$$

$[\phi_1]$ $\textsc{CandLbl}_{\text{struct}}(\texttt{FabFour}, \texttt{novel})$

$\Rightarrow$ $\textsc{Lbl}(\texttt{FabFour}, \texttt{novel})$

$[\phi_2]$ $\textsc{CandRel}_{\text{pat}}(\texttt{Beatles}, \texttt{AlbumArtist}, \texttt{AbbeyRoad})$

$\Rightarrow$ $\textsc{Rel}(\texttt{Beatles}, \texttt{AlbumArtist}, \texttt{AbbeyRoad})$

$[\phi_3]$ $\textsc{SameEnt}(\texttt{Beatles}, \texttt{FabFour})$

$\wedge\ \textsc{Lbl}(\texttt{Beatles}, \texttt{musician})$

$\Rightarrow \textsc{Lbl}(\texttt{FabFour}, \texttt{musician})$

$[\phi_4]$ $\textsc{Dom}(\texttt{AlbumArtist}, \texttt{musician})$

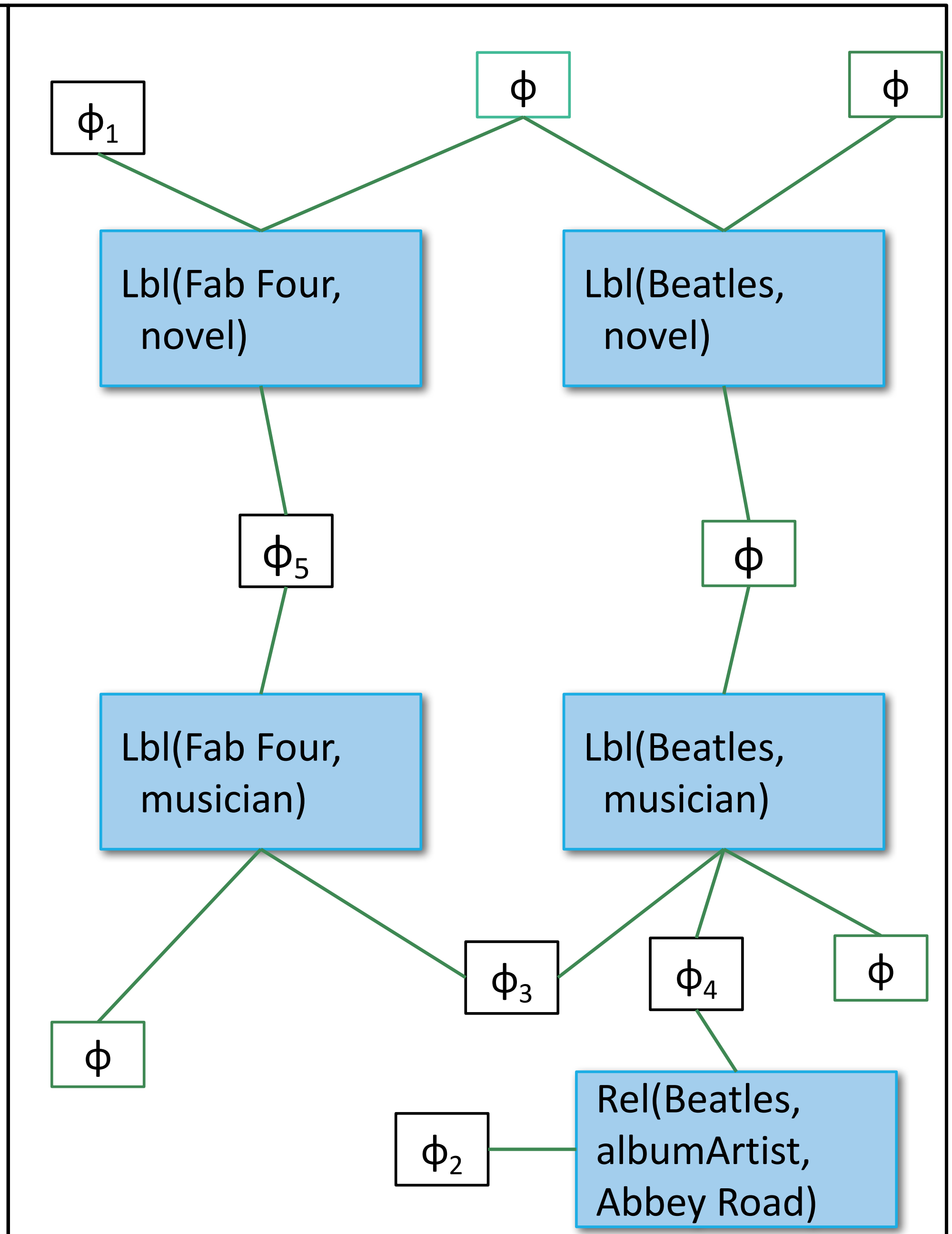
$\wedge\ \textsc{Rel}(\texttt{Beatles}, \texttt{AlbumArtist}, \texttt{AbbeyRoad})$

$\Rightarrow \textsc{Lbl}(\texttt{Beatles}, \texttt{musician})$

$[\phi_5]$ $\textsc{Mut}(\texttt{musician}, \texttt{novel})$

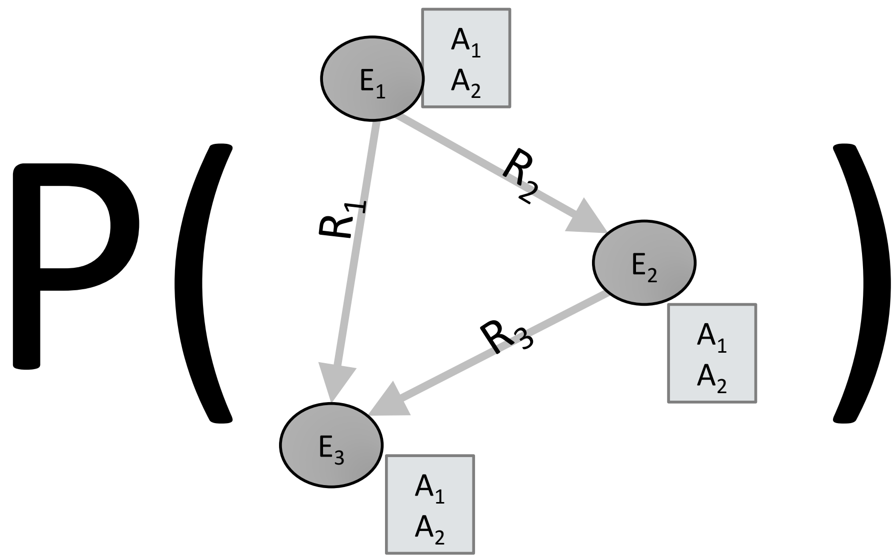
$\wedge\ \textsc{Lbl}(\texttt{FabFour}, \texttt{musican})$

$\Rightarrow \neg\textsc{Lbl}(\texttt{FabFour}, \texttt{novel})$

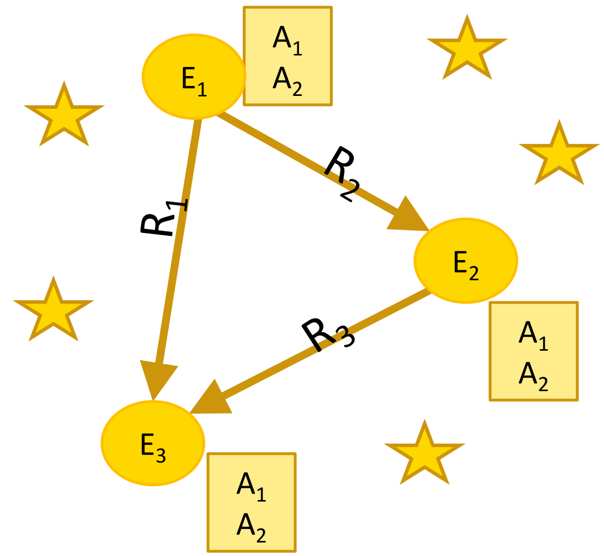


How do we get a knowledge graph?

Have: P(KG) forall 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

	AUC	F1
Extract	.873	.828
Rules	.765	.673
MLN (Jiang, 12)	.899	.836
PSL (Pujara, 13)	.904	.853

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

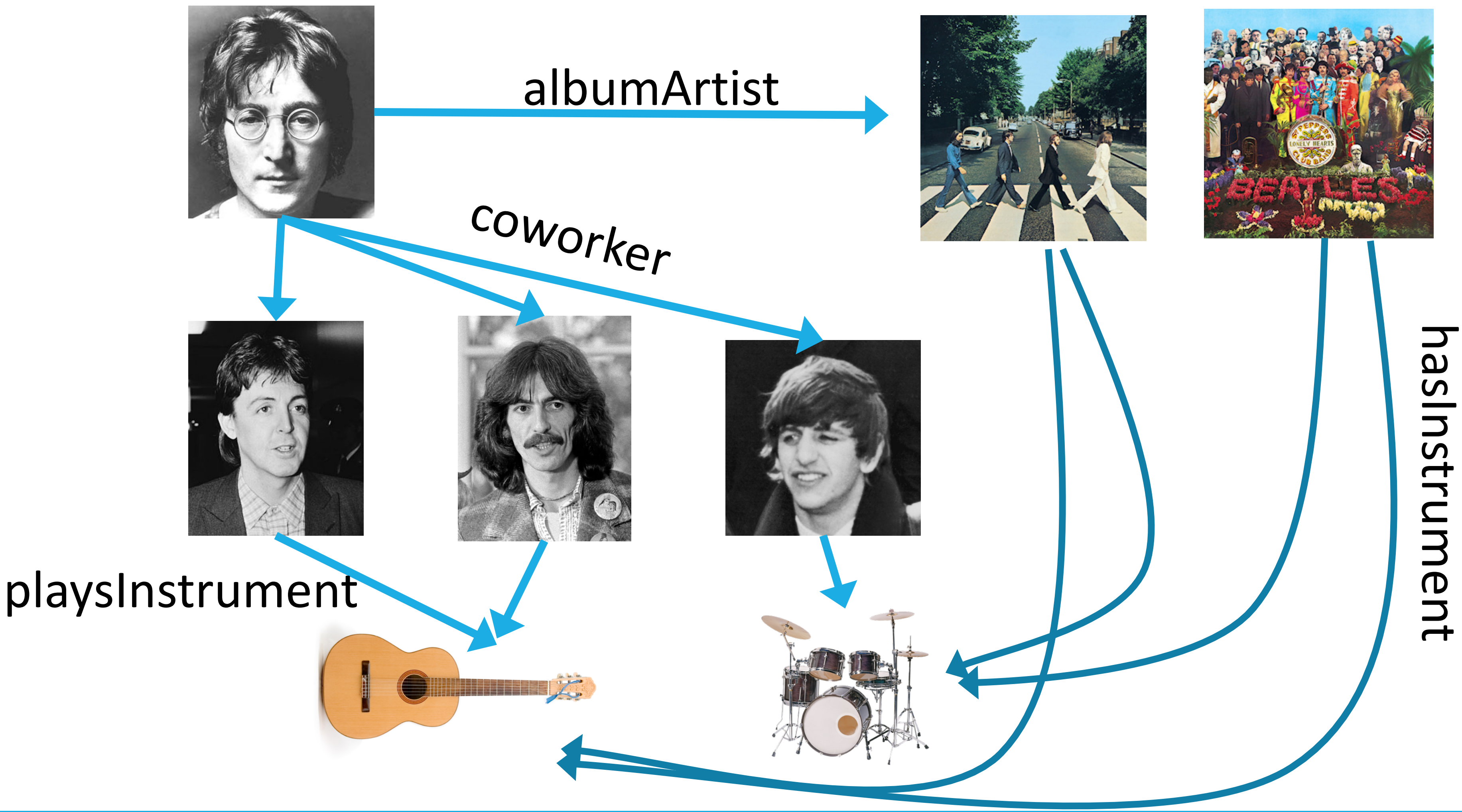
- Given: a query of an **entity** and **relation**
- Starting at the entity, **randomly walk** the KG
- Random walk ends when reaching an appropriate **goal**
- Learned **parameters** bias choices in the random walk
- Output **relative probabilities** of goal states

Random Walk Illustration

Query: R(Lennon, PlaysInstrument, ?)

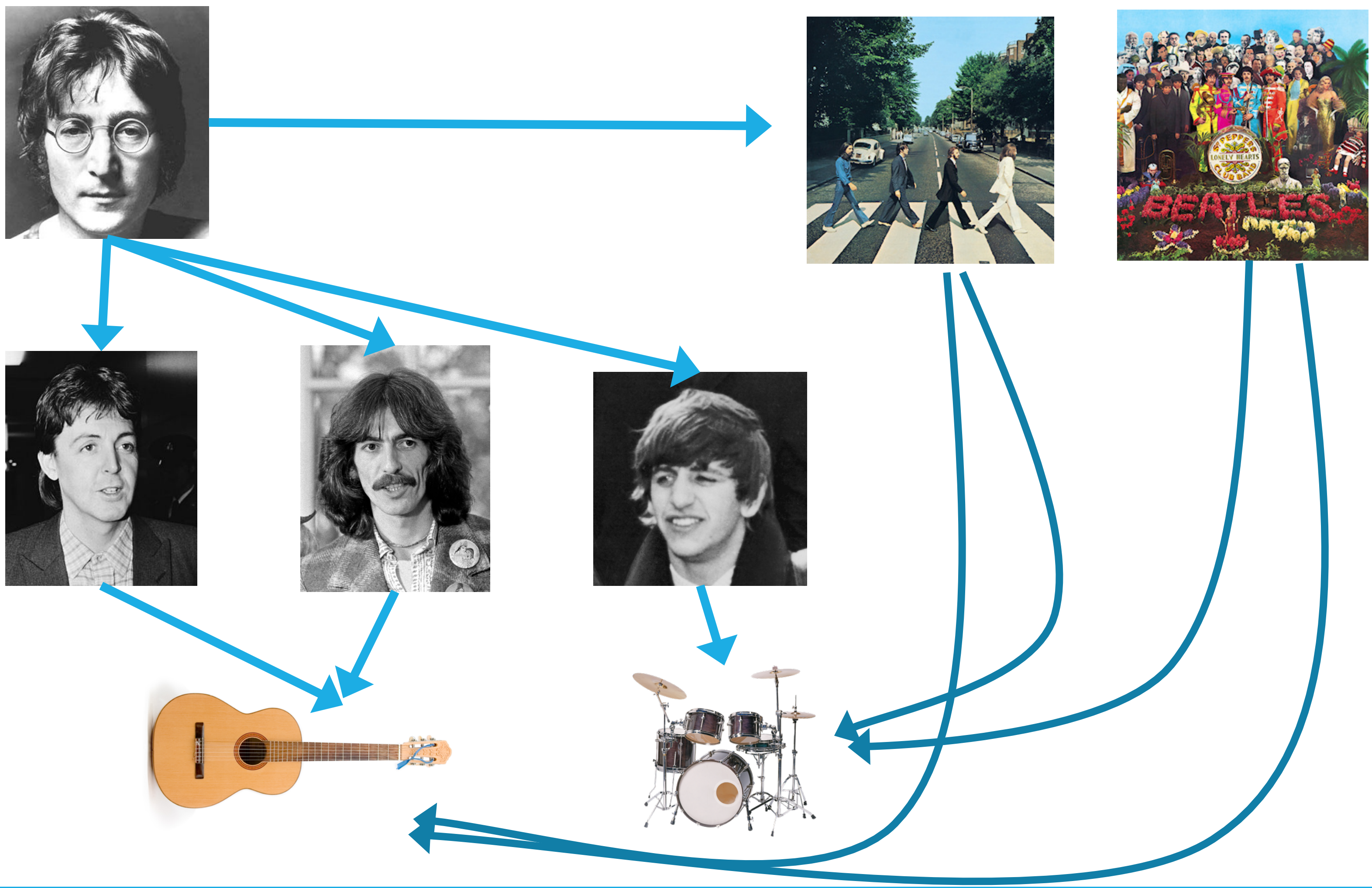
Random Walk Illustration

Query: R(Lennon, PlaysInstrument, ?)



Random Walk Illustration

Query: R(Lennon, PlaysInstrument, ?)

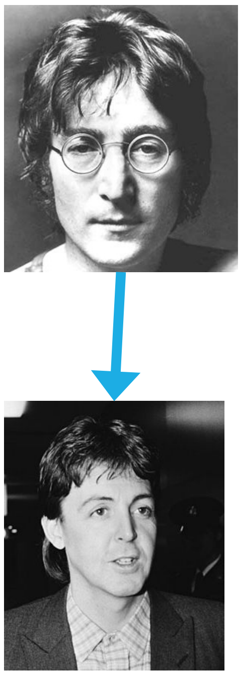


Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)

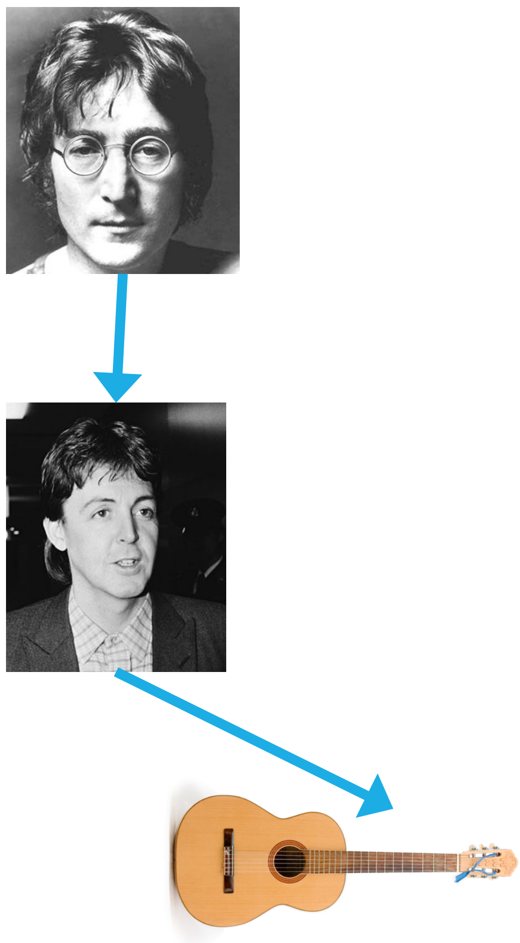
Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)



Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)



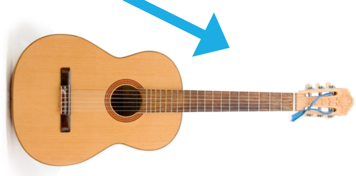
Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)

Path

Weight of path

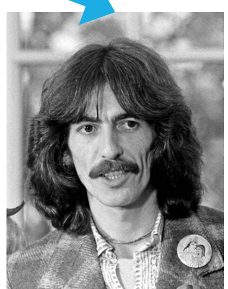
$$\mathbf{P(Q|\pi=\langle coworker,playsInstrument\rangle)\ W_{\pi}}$$



Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)

P(Q|π=<coworker,playsInstrument>) W_{π}



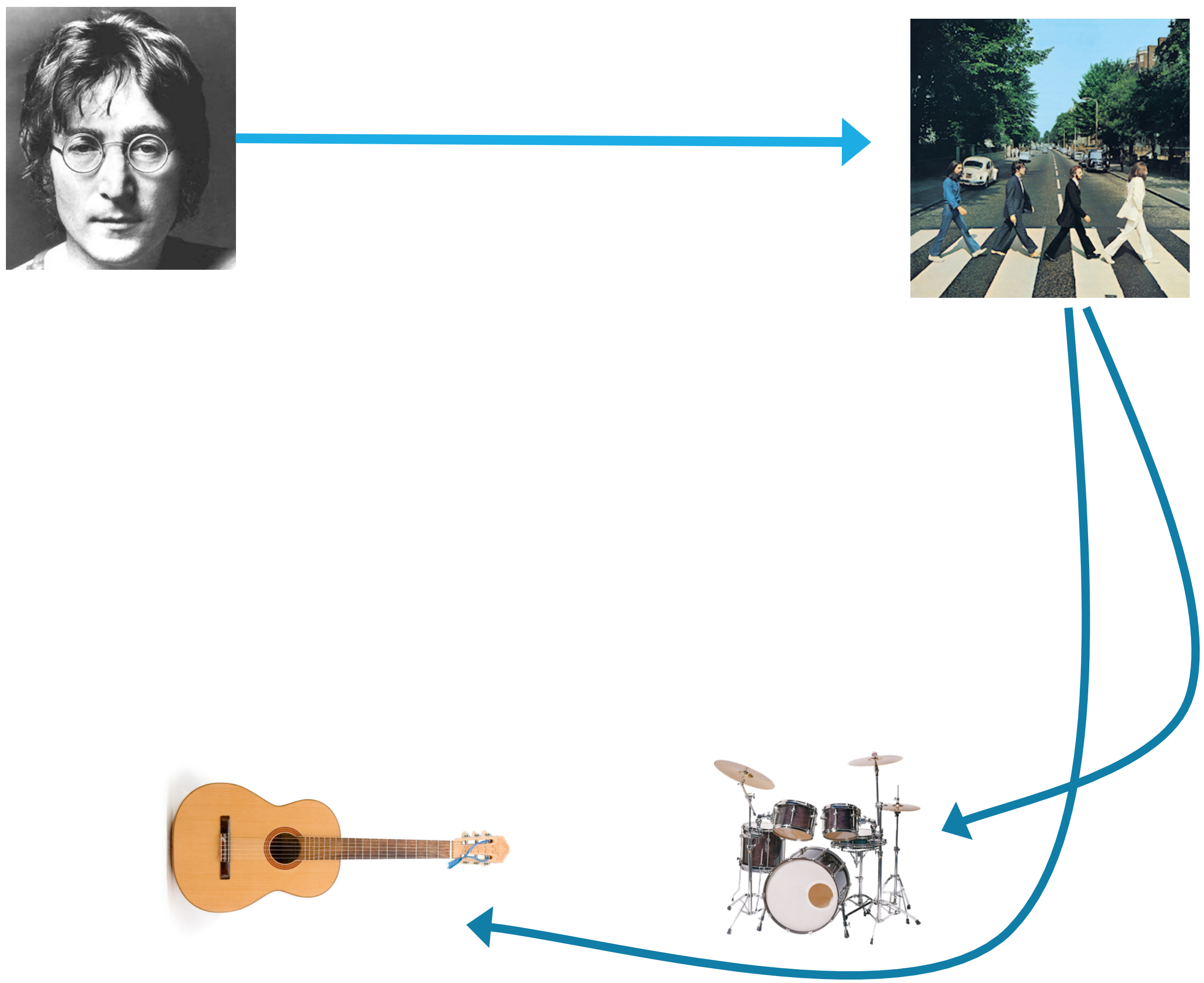
Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)

$\mathbf{P(Q|\pi=\langle coworker,playsInstrument\rangle)\ W_{\pi}}$

Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)



Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)

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

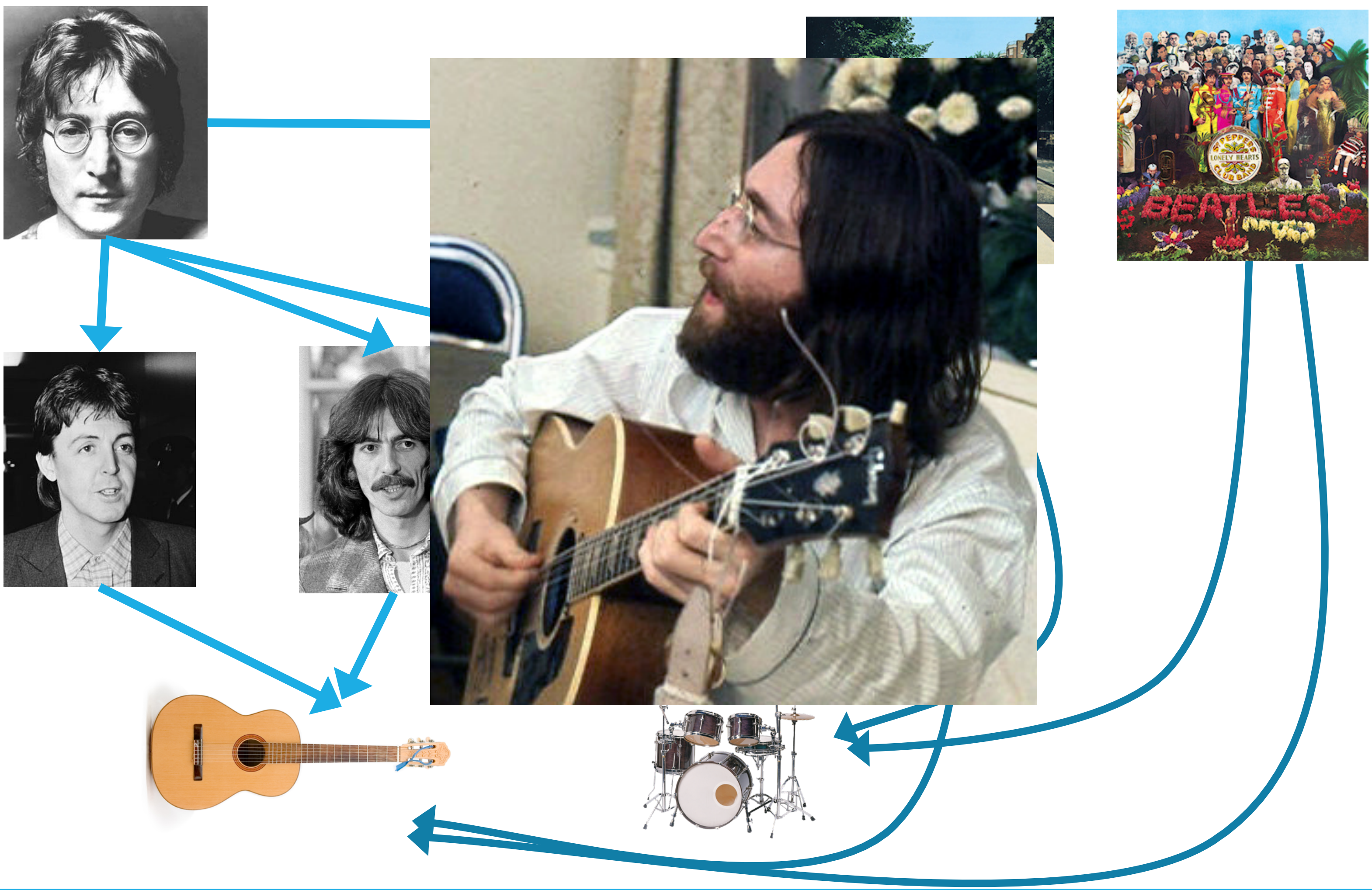
Random Walk Illustration

Query Q: R(Lennon, PlaysInstrument, ?)

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

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

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

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

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

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

R(,AlbumArtist,J)
R(J,HasInstrument,K)

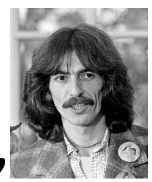
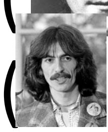
Unbound variables in proof tree!

ProPPR-ized PRA example

Query Q: R(Lennon, PlaysInstrument, ?)

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

R(,AlbumArtist,J)
R(J,HasInstrument,K)

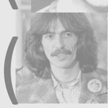
R(,Coworker,)
R(,PlaysInstrument,Y)

ProPPR-ized PRA example

Query Q: R(Lennon, PlaysInstrument, ?)

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

R( ,AlbumArtist,J)
R(J,HasInstrument,K)

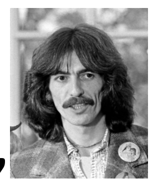
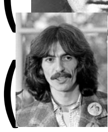
R( ,Coworker, )
R( ,PlaysInstrument,Y)

ProPPR-ized PRA example

Query Q: R(Lennon, PlaysInstrument, ?)

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

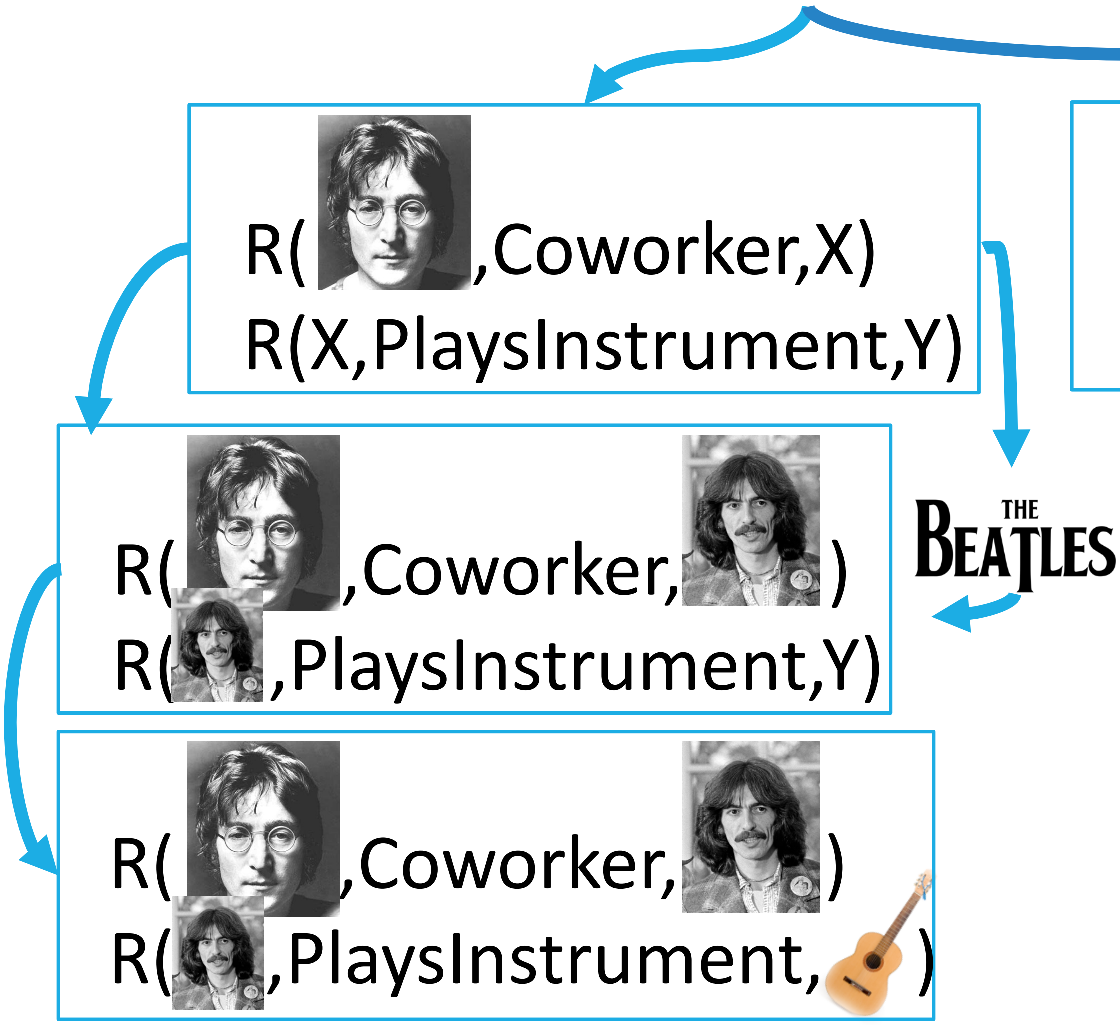
R( ,AlbumArtist,J)
R(J,HasInstrument,K)

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

THE BEATLES

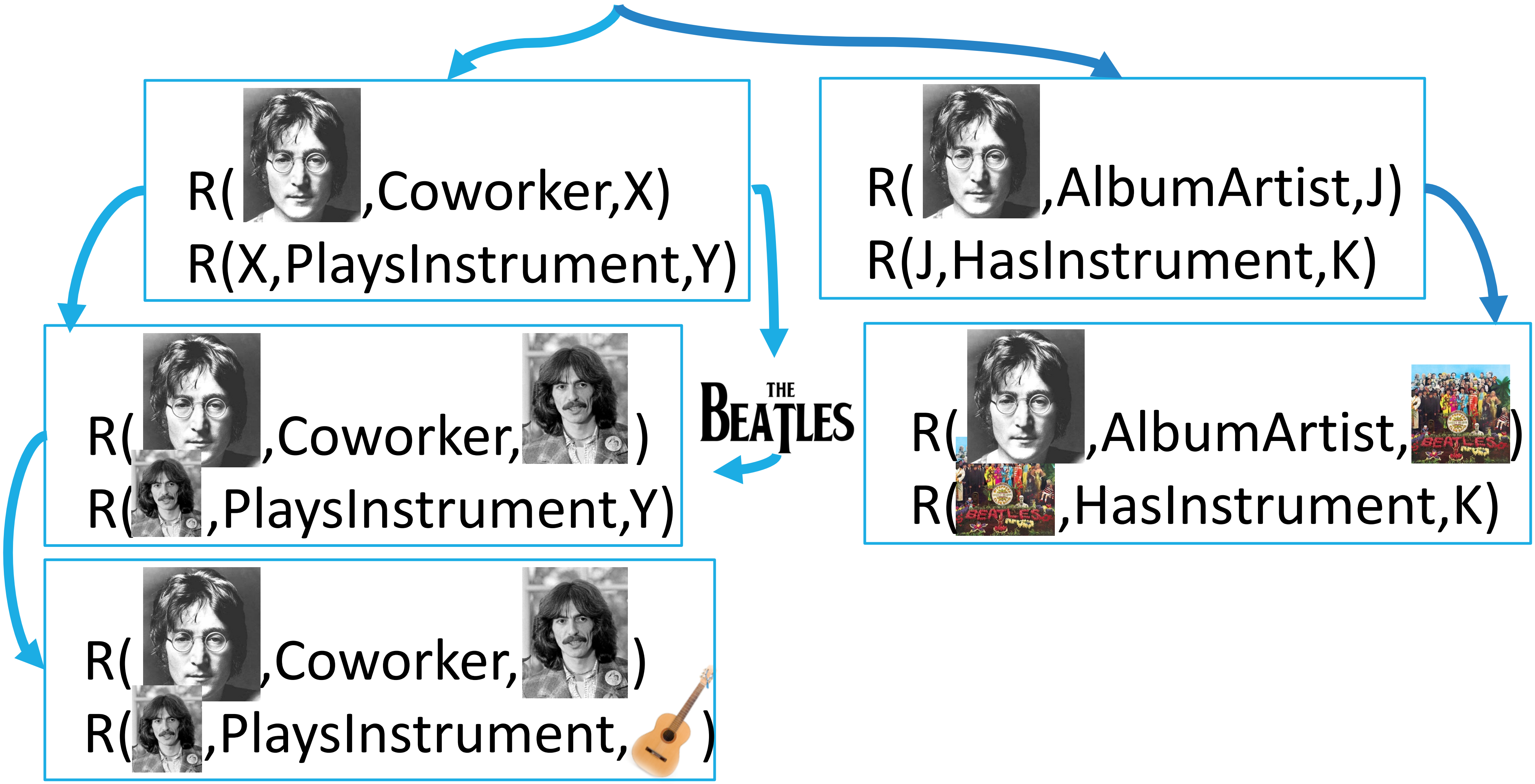
ProPPR-ized PRA example

Query Q: R(Lennon, PlaysInstrument, ?)



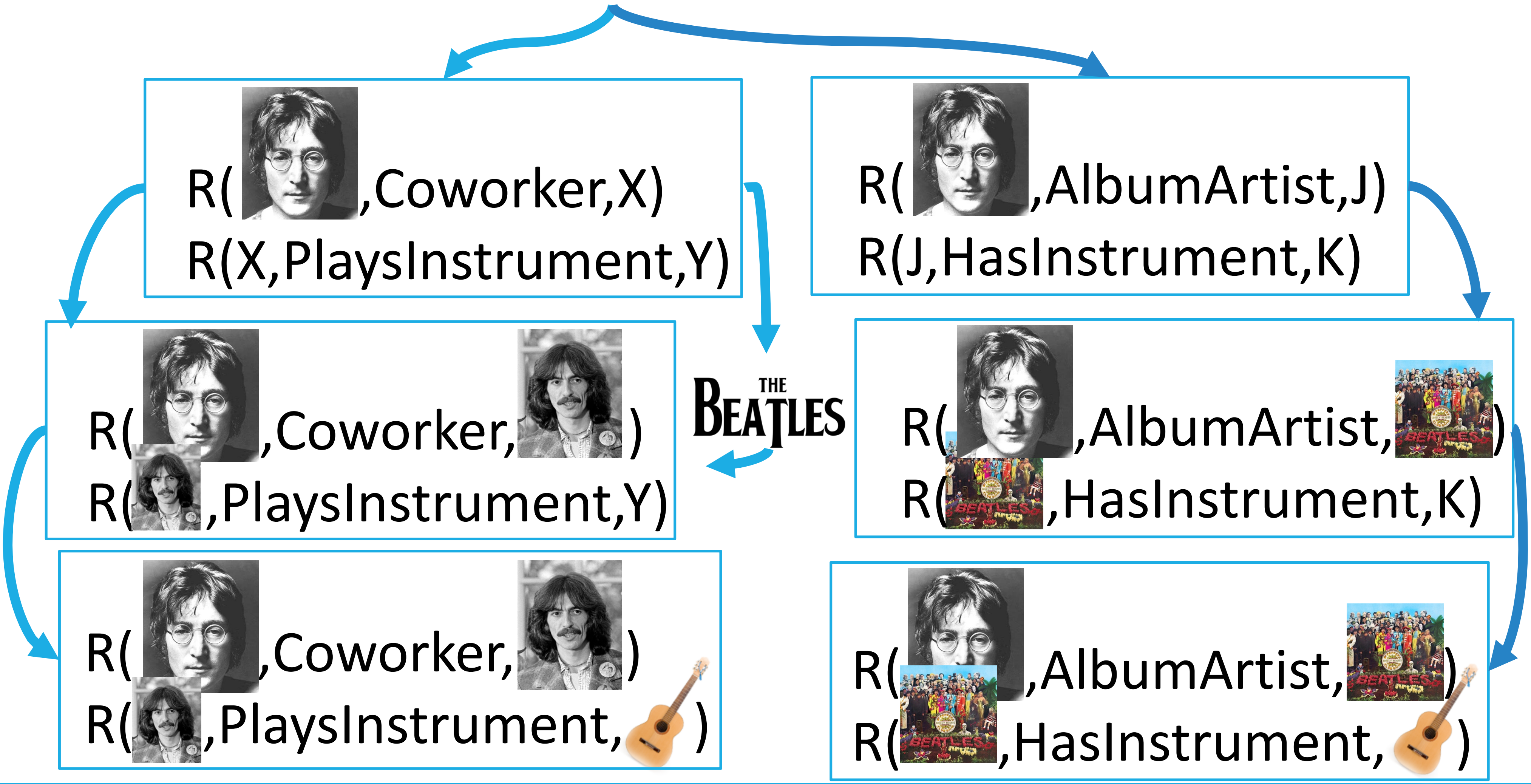
ProPPR-ized PRA example

Query Q: R(Lennon, PlaysInstrument, ?)



ProPPR-ized PRA example

Query Q: R(Lennon, PlaysInstrument, ?)



ProPPR-ized PRA example

Query Q: R(Lennon, PlaysInstrument, ?)

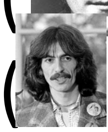
R(,Cowor

R(X,PlaysInstru

AlbumArtist,J)

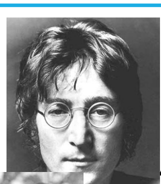
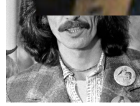
nstrument,K)

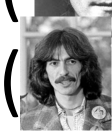
R(,Coworker,

R(,PlaysInstrume

,AlbumArtist,)

HasInstrument,K)

R(,Coworker,)

R(,PlaysInstrument,)

R(,AlbumArtist,)

R(,HasInstrument,)

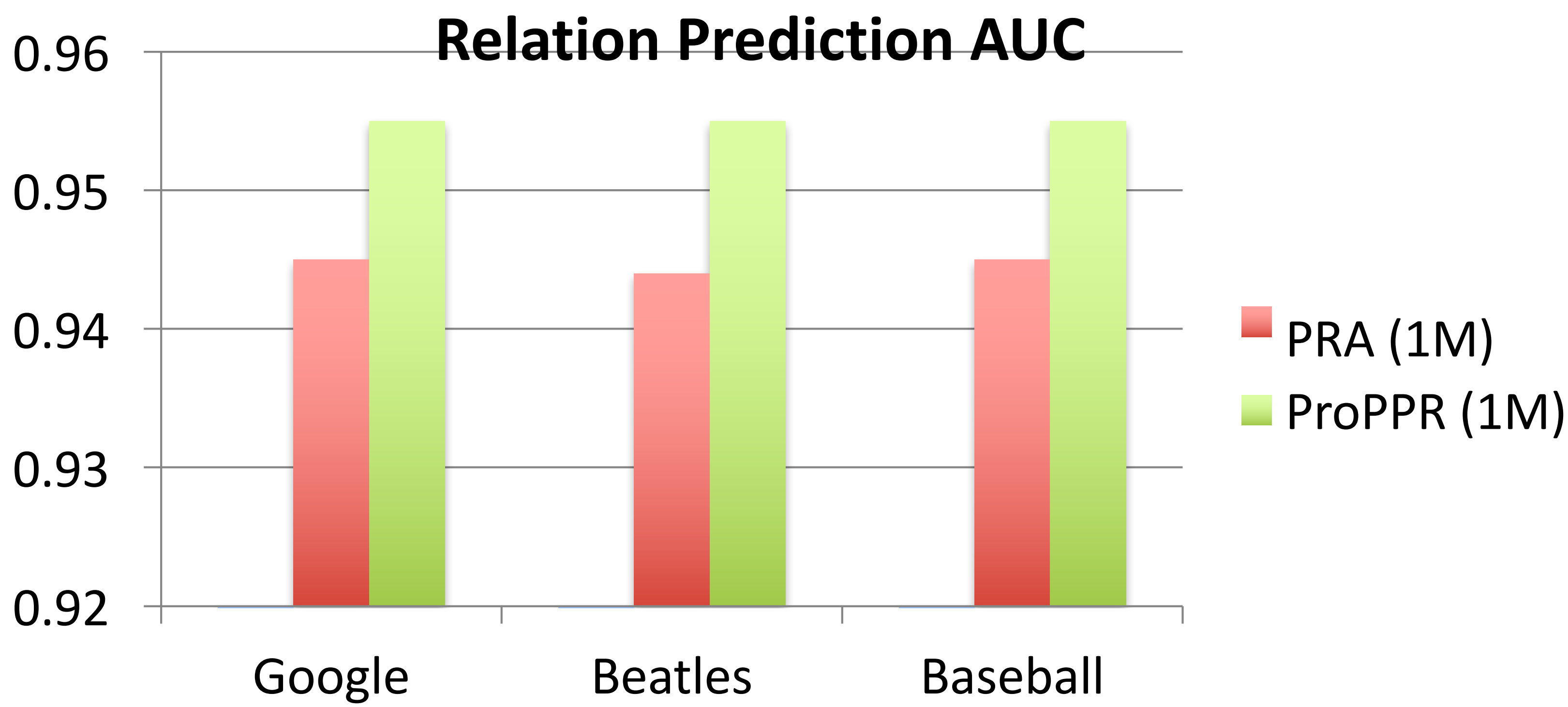
ProPPR in a nutshell

$$\min_{\mathbf{w}} -\left(\sum_{k\in+}\log \mathbf{p}_{\nu_0}[u_+^k] + \sum_{k\in-}\log(1-\mathbf{p}_{\nu_0}[u_-^k]\right) + \mu||\mathbf{w}||_2^2$$

- Input: queries, positive answers, negative answers
- Goal: $\mathbf{p}_{\nu_0}[u_+^k] \geq \mathbf{p}_{\nu_0}[u_-^k]$ (page rank from RW)
- Learn: random walk weights
- Train via stochastic gradient descent

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

GRAPHICAL MODELS

- Possible facts in KG are variables
- Logical rules relate facts
- Probability $\propto$ satisfied rules
- Universally-quantified

RANDOM WALK METHODS

- Possible facts posed as queries
- Random walks of the KG constitute “proofs”
- Probability $\propto$ path lengths/transitions
- Locally grounded

Embedding-Based Techniques

MATRICES, TENSORS, AND NEURAL NETWORKS

Probabilistic Models: Downsides

Embeddings

Limitation to Logical Relations

- Representation restricted by manual design
  - Clustering? Assymetric implications?
  - Information flows through these relations
- Difficult to generalize to unseen entities/relations

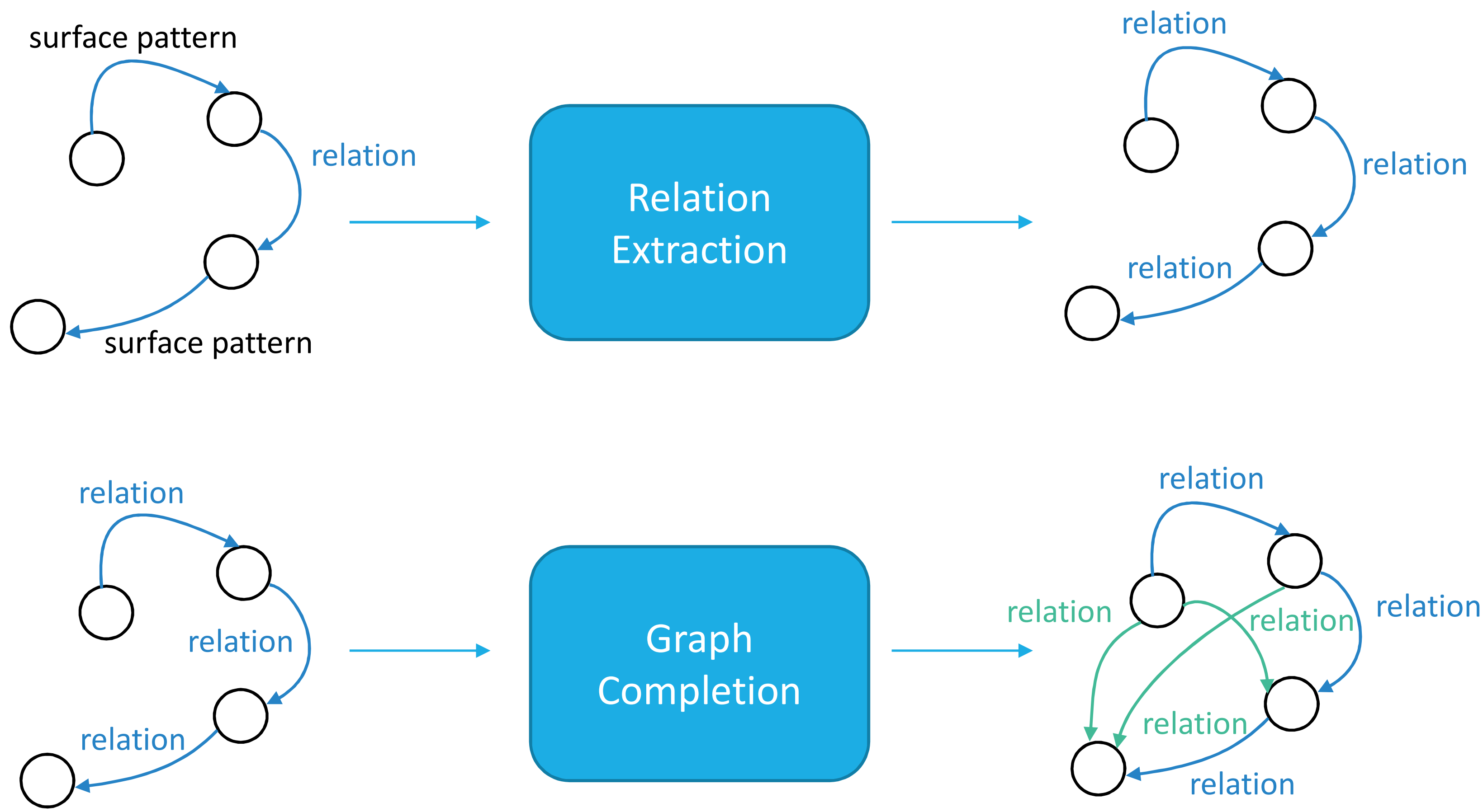
- Everything as dense vectors
- Can capture many relations
- Learned from data

Computational Complexity of Algorithms

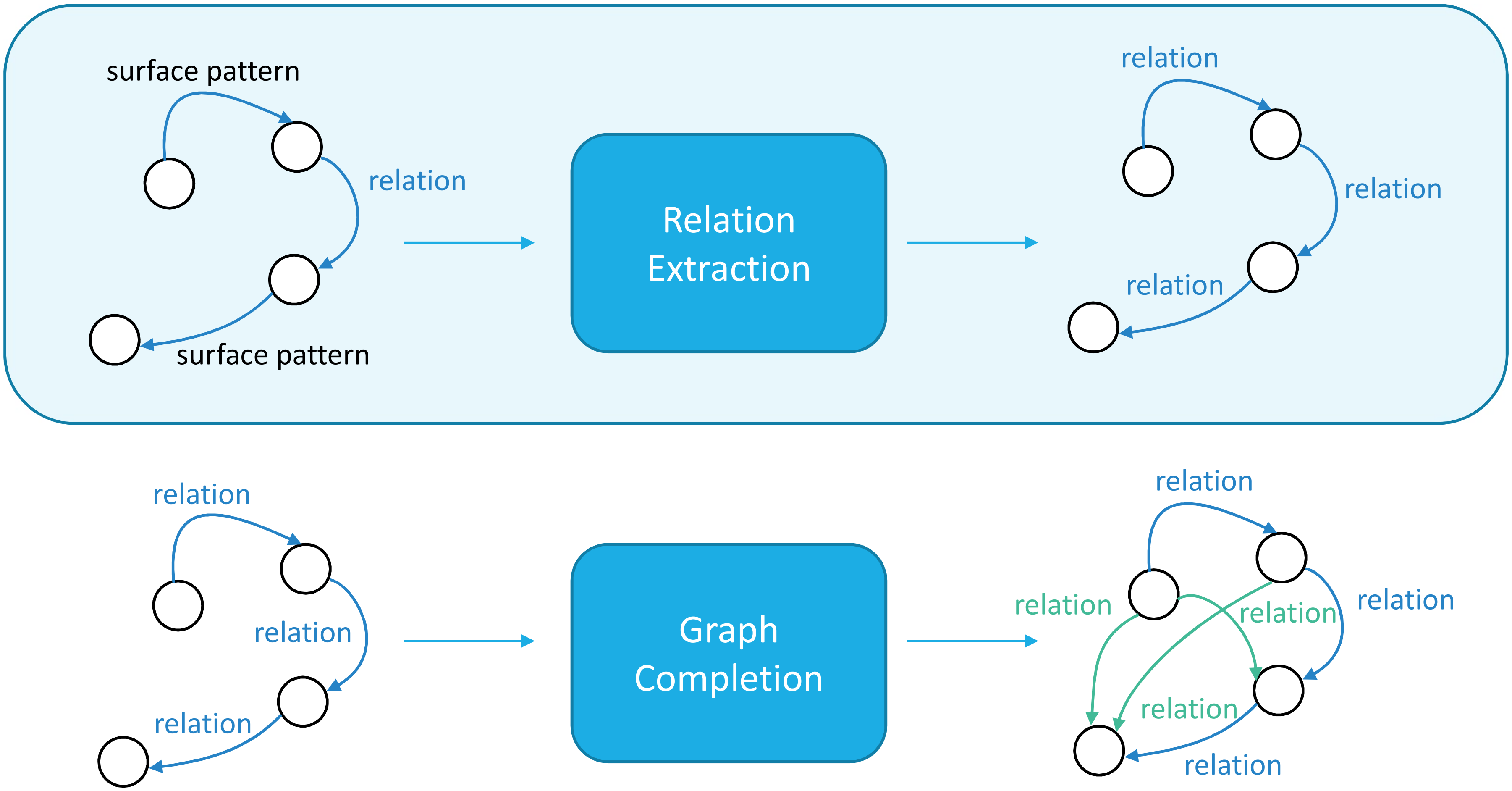
- Complexity depends on explicit dimensionality
  - Often NP-Hard, in size of data
  - More rules, more expensive inference
- Query-time inference is sometimes NP-Hard
- Not trivial to parallelize, or use GPUs

- Complexity depends on latent dimensions
- Learning using stochastic gradient, back-propagation
- Querying is often cheap
- GPU-parallelism friendly

Two Related Tasks

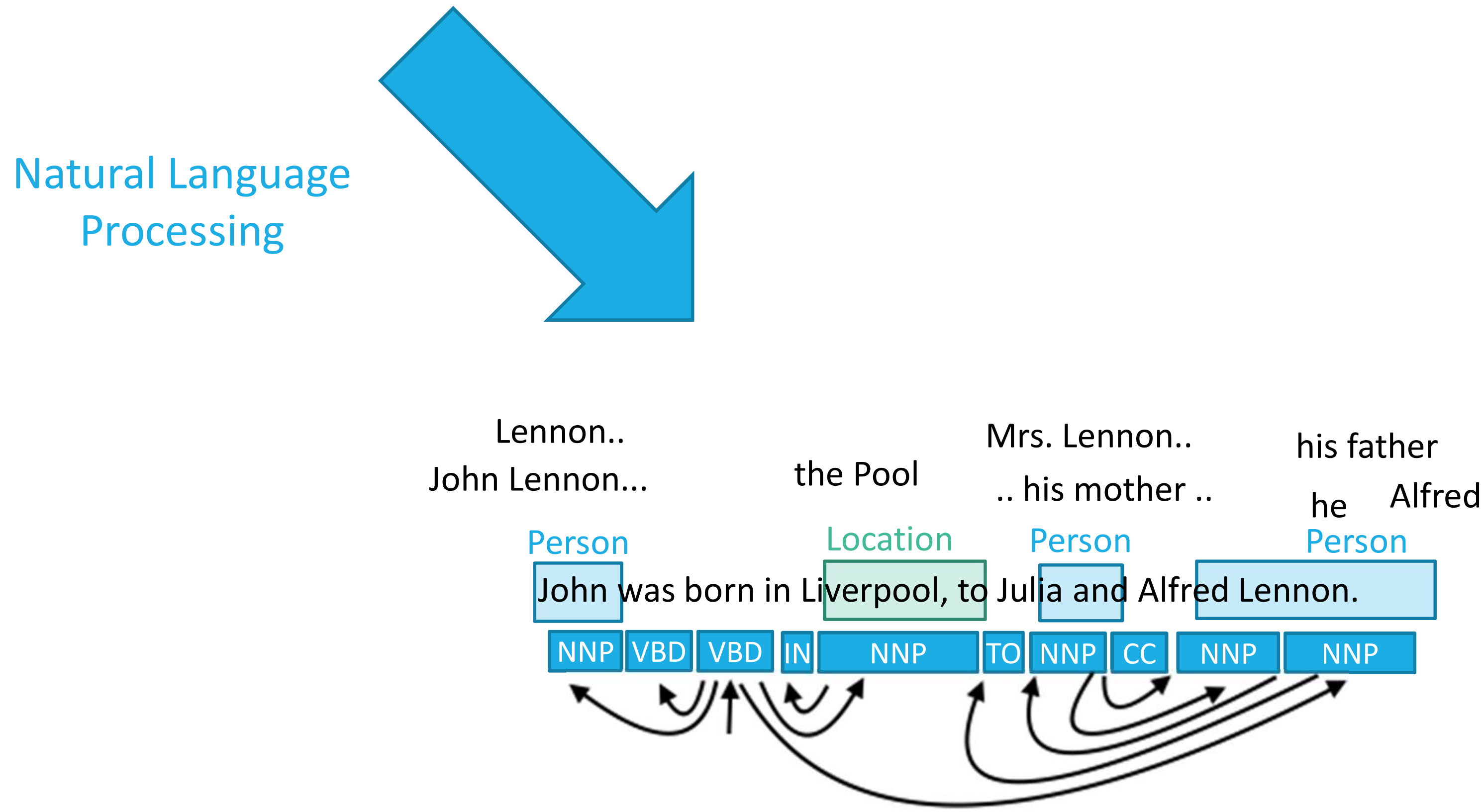


Two Related Tasks

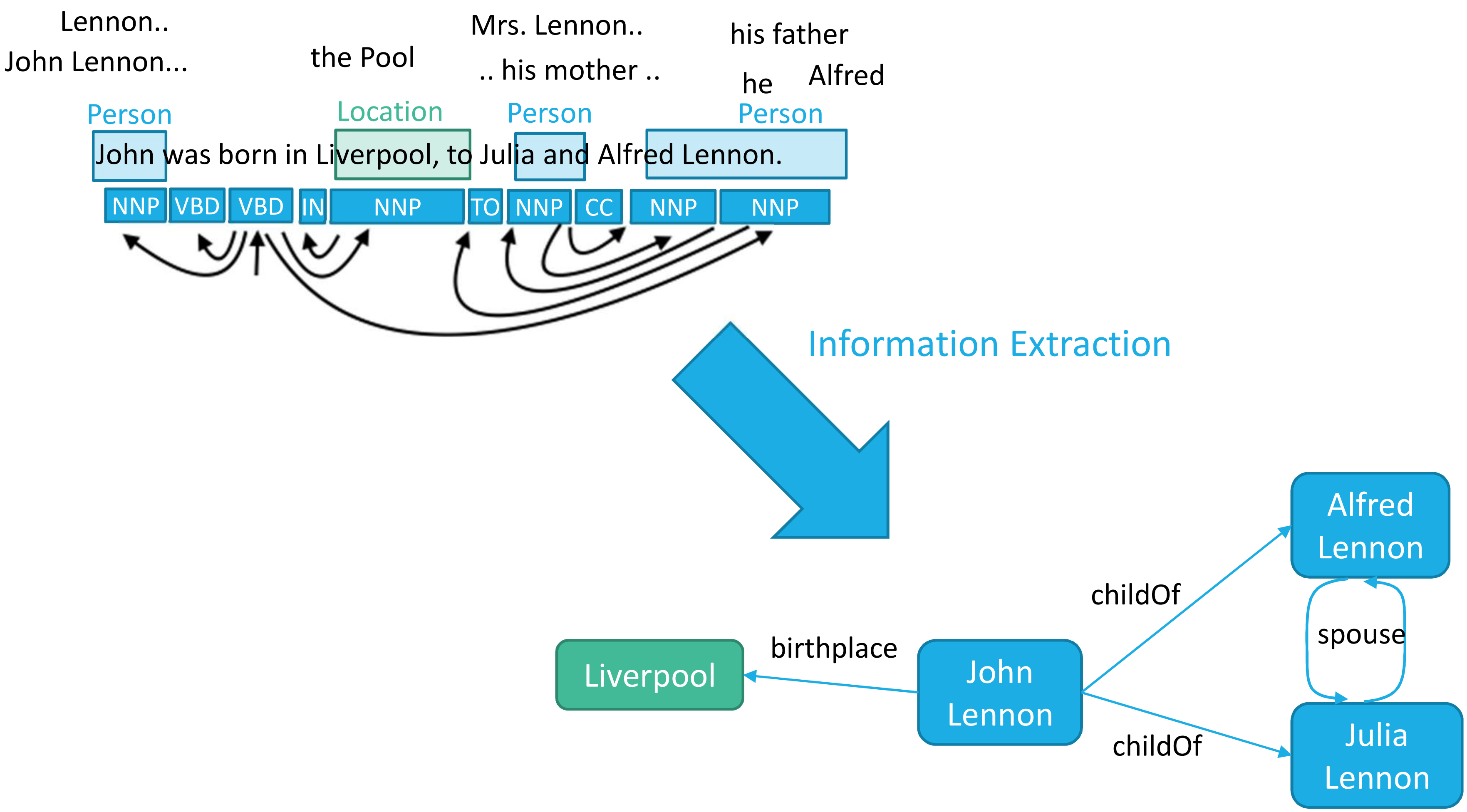


What is NLP?

John was born in Liverpool, to Julia and Alfred Lennon.

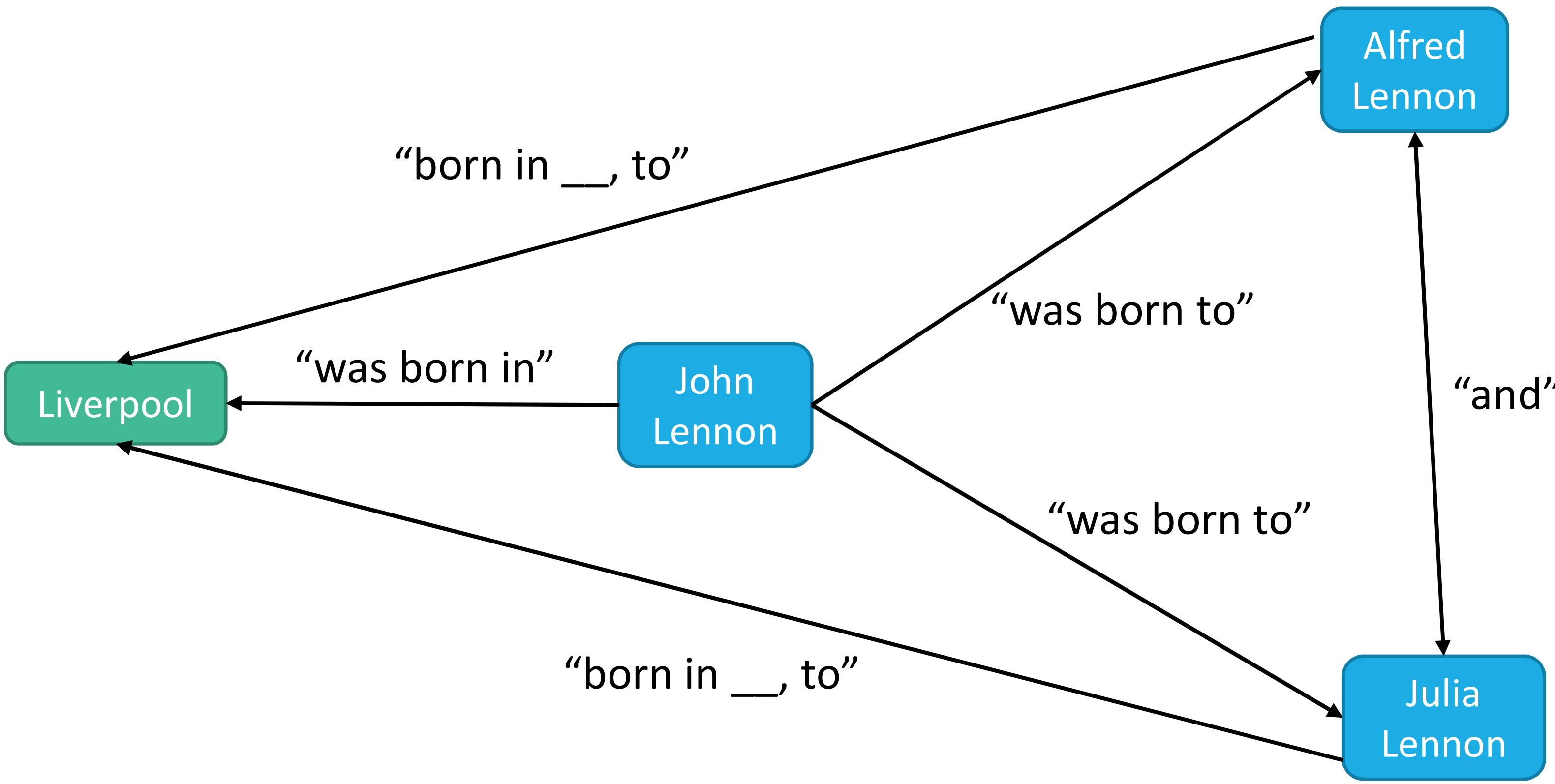


What is Information Extraction?



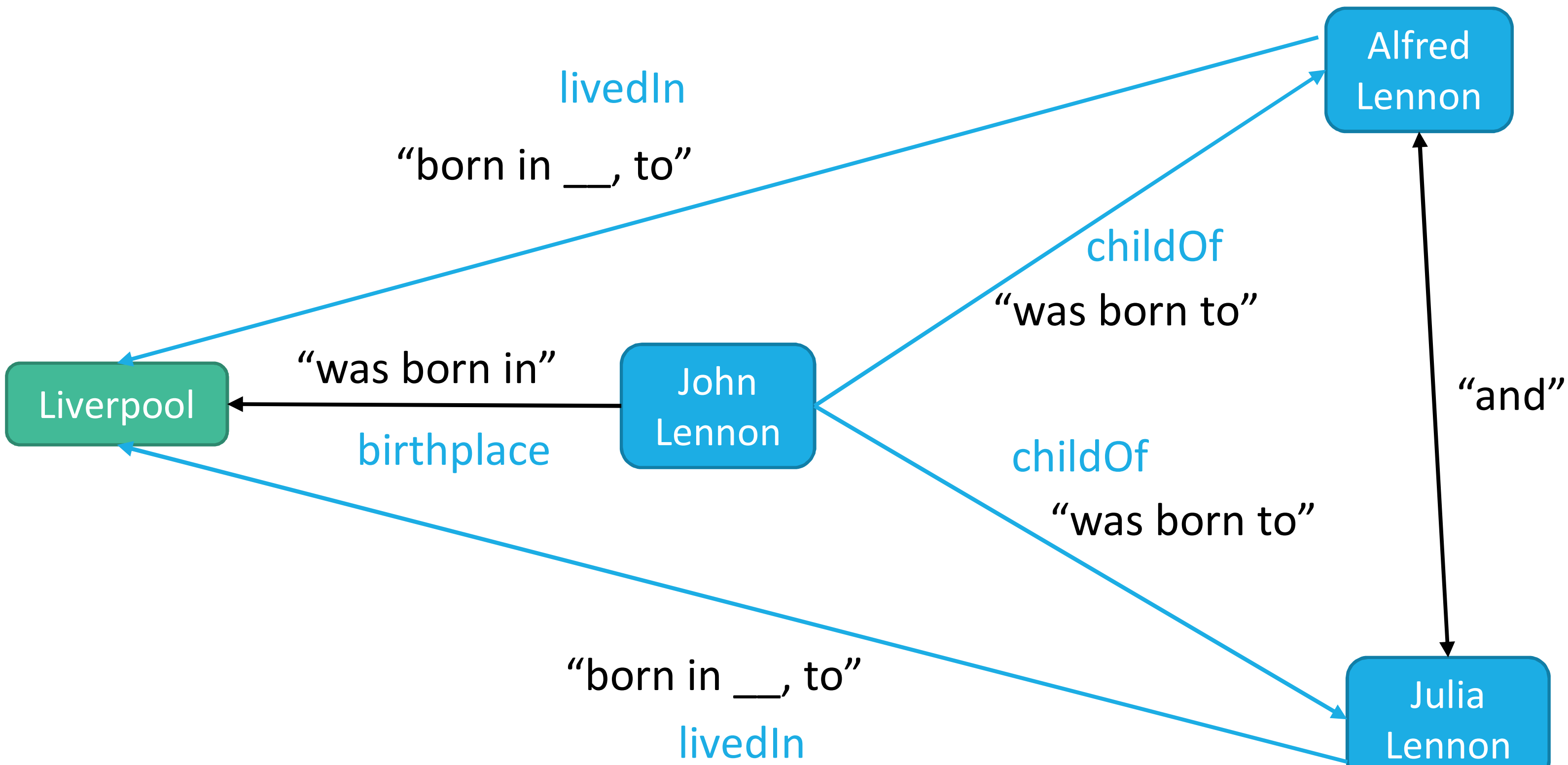
Relation Extraction From Text

John was born in Liverpool, to Julia and Alfred Lennon.



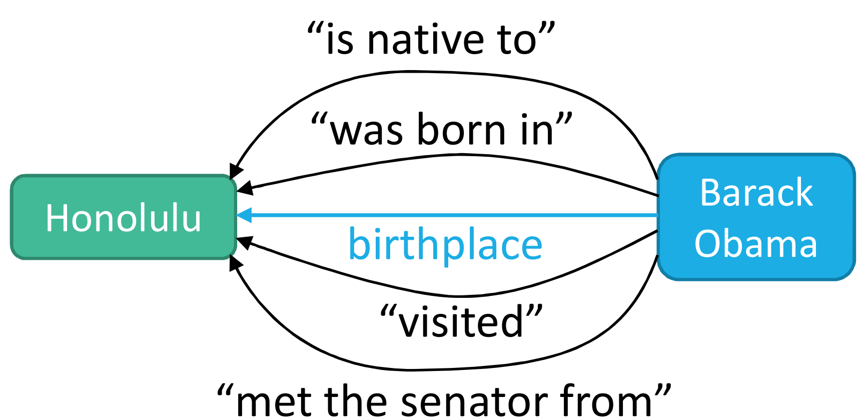
Relation Extraction From Text

John was born in Liverpool, to Julia and Alfred Lennon.



“Distant” Supervision

No direct supervision gives us this information.
Supervised: Too expensive to label sentences
Rule-based: Too much variety in language
Both only work for a small set of relations, i.e. 10s, not 100s



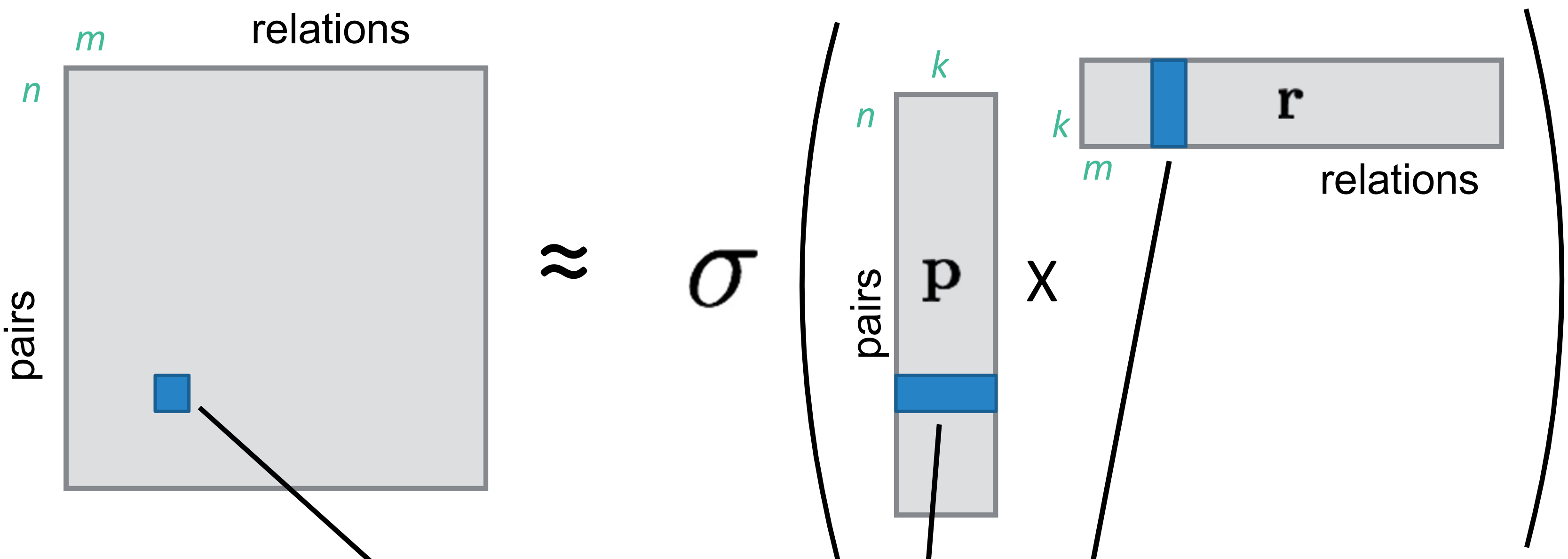
Relation Extraction as a Matrix

John was born in Liverpool, to Julia and Alfred Lennon.

Entity Pairs

	was born in <-nsubjpass-born<-nmod:in-	was born to	and	birthplace(X,Y)	spouse(X,Y)
John Lennon, Liverpool	1			?	
John Lennon, Julia Lennon		1			
John Lennon, Alfred Lennon		1			
Julia Lennon, Alfred Lennon			1		?
Barack Obama, Hawaii	1			1	
Barack Obama, Michelle Obama			1		1

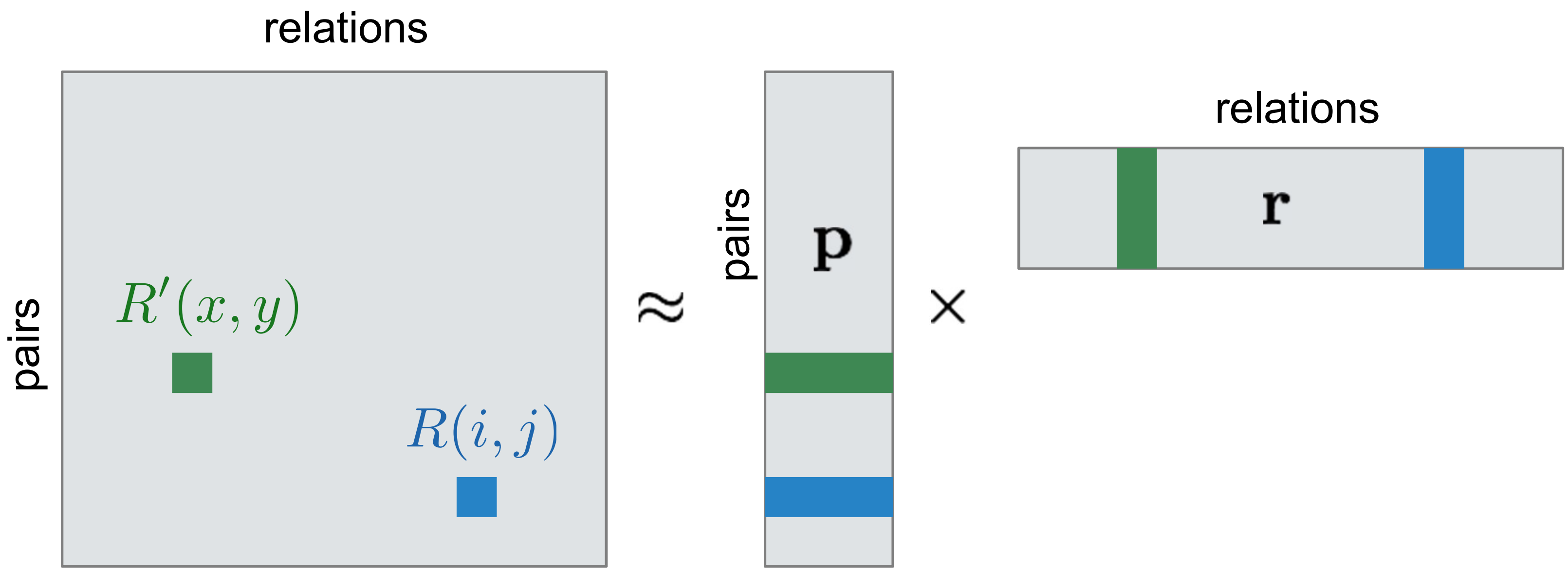
Matrix Factorization



bornIn(John,Liverpool)

$$P(R(i,j)) = \sigma(\mathbf{p}_{i,j} \cdot \mathbf{r}_R)$$

Training: Stochastic Updates



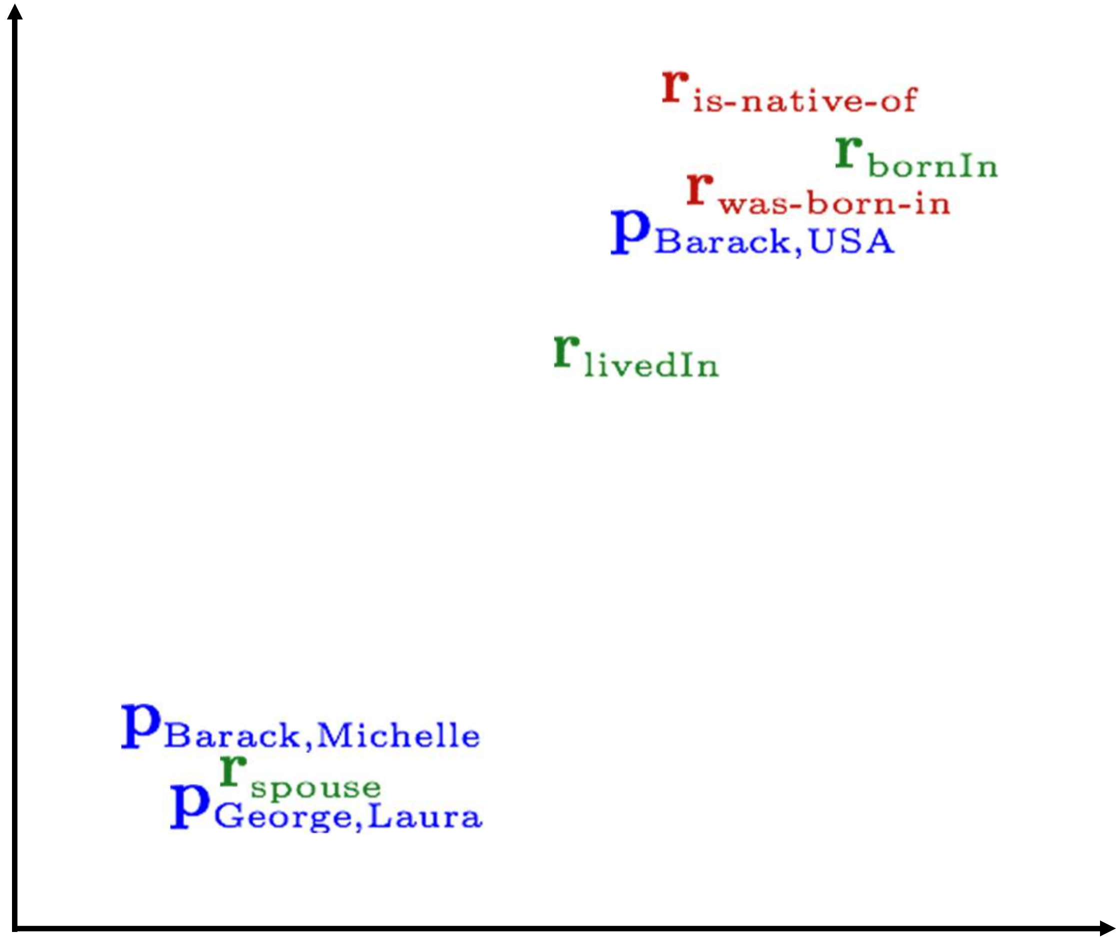
Pick an observed cell, $R(i,j)$:

- Update $\mathbf{p}_{ij}$ & $\mathbf{r}_R$ such that $R(i,j)$ is higher

Pick any random cell, assume it is negative:

- Update $\mathbf{p}_{xy}$ & $\mathbf{r}_{R'}$ such that $R'(x,y)$ is lower

Relation Embeddings



Embeddings ~ Logical Relations

Relation Embeddings, w

- Similar embedding for 2 relations denote they are paraphrases
  - is married to, spouseOf(X,Y), /person/spouse
- One embedding can be contained by another
  - w(topEmployeeOf) ⊂ w(employeeOf)
  - topEmployeeOf(X,Y) → employeeOf(X,Y)
- Can capture logical patterns, without needing to specify them!

Entity Pair Embeddings, v

Similar entity pairs denote similar relations between them

Entity pairs may describe multiple "relations"

independent foundedBy and employeeOf relations

Similar Embeddings

similar underlying embedding

similar embedding

	X **own percentage of** Y	X **buy stake in** Y
Time, Inc Amer. Tel. and Comm.	1	1
Volvo Scania A.B.		1
Campeau Federated Dept Stores		
Apple HP		

Successfully predicts “Volvo owns percentage of Scania A.B.” from “Volvo bought a stake in Scania A.B.”

Implications

X historian at Y → X professor at Y

(Freeman,Harvard)
→ (Boyle,OhioState)

	X **professor at** Y	X **historian at** Y
Kevin Boyle Ohio State		1
R. Freeman Harvard	1	

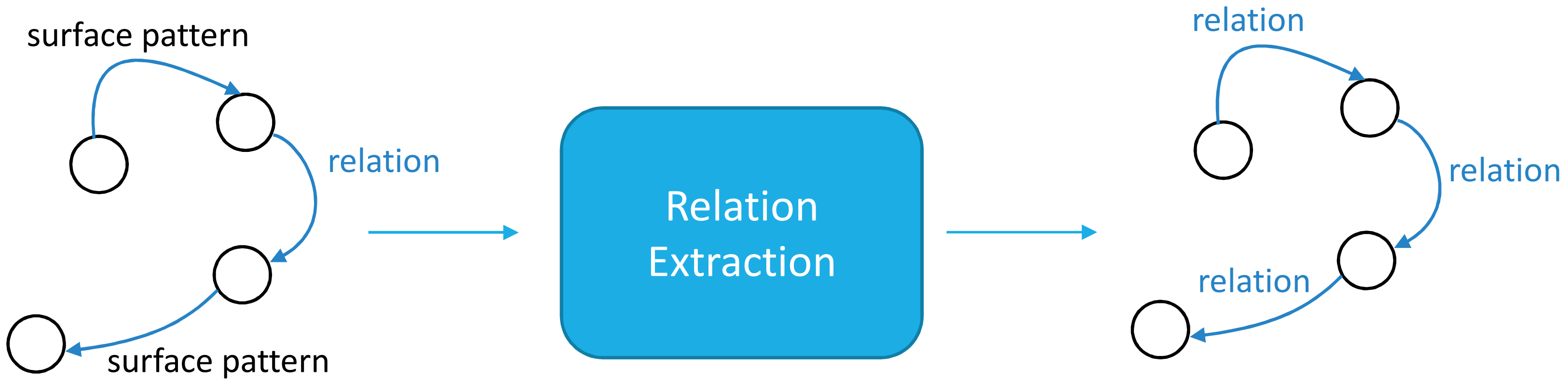
Learns asymmetric entailment:

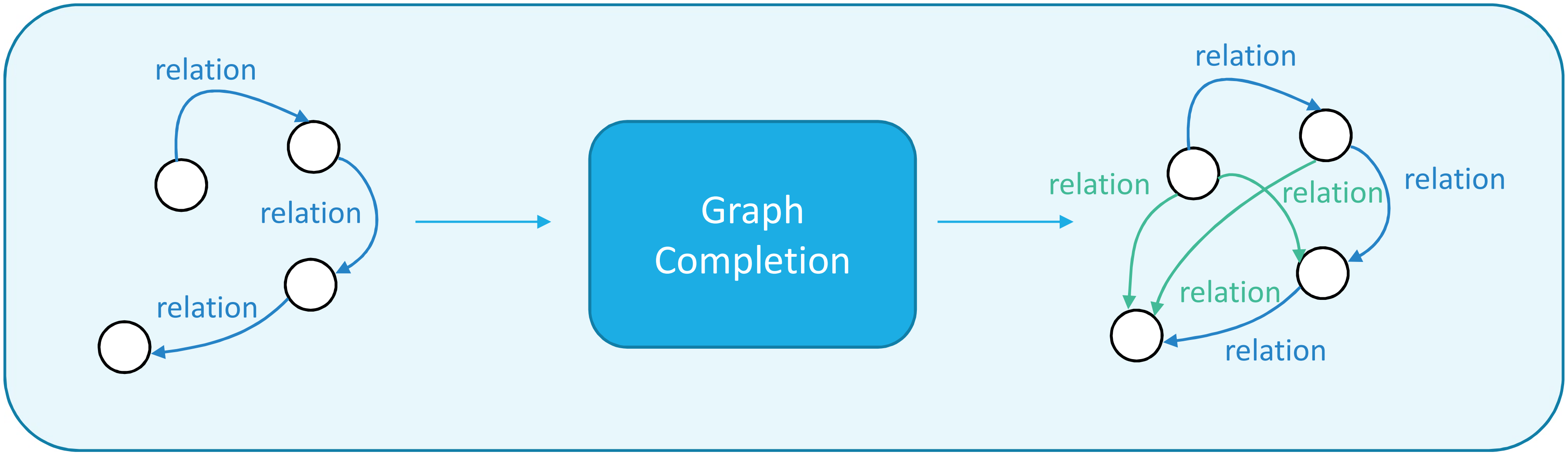
PER historian at UNIV $\rightarrow$ PER professor at UNIV

But,

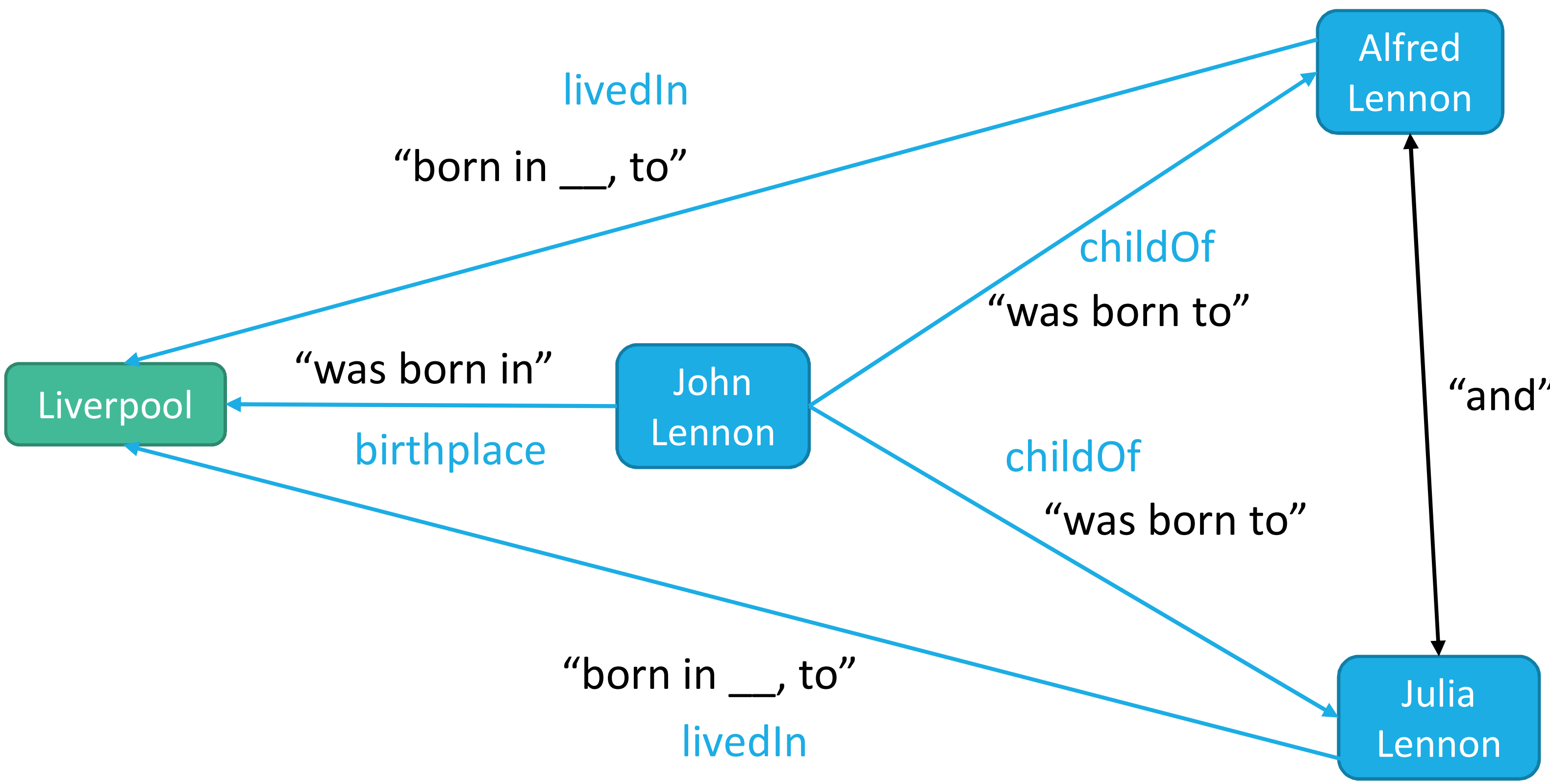
PER professor at UNIV $\not\rightarrow$ PER historian at UNIV

Two Related Tasks

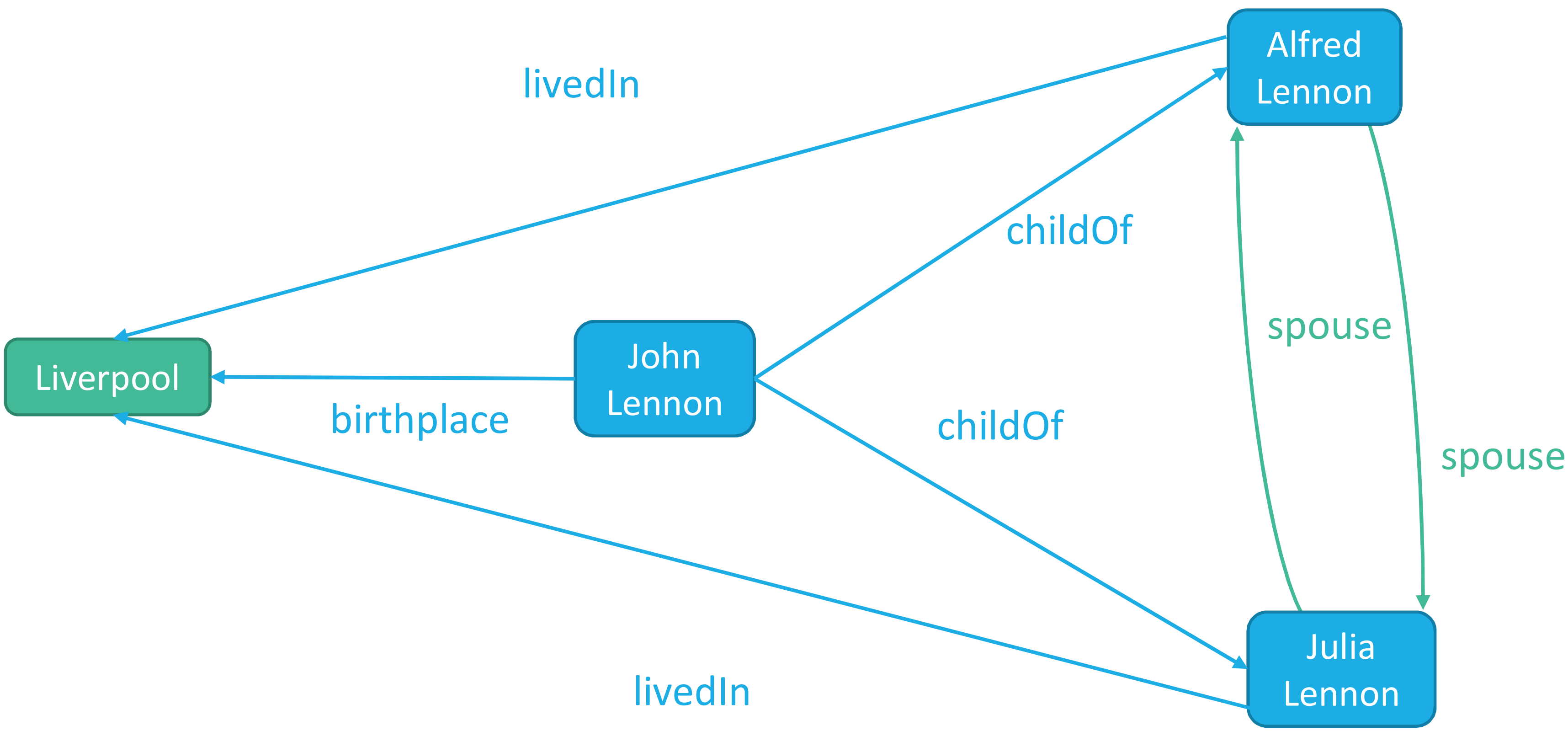




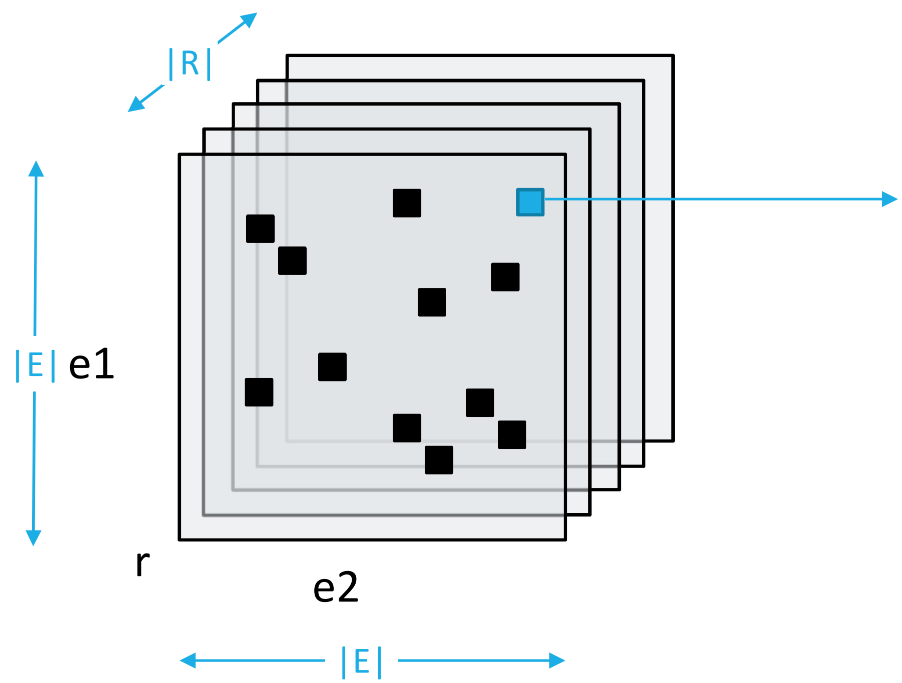
Graph Completion



Graph Completion

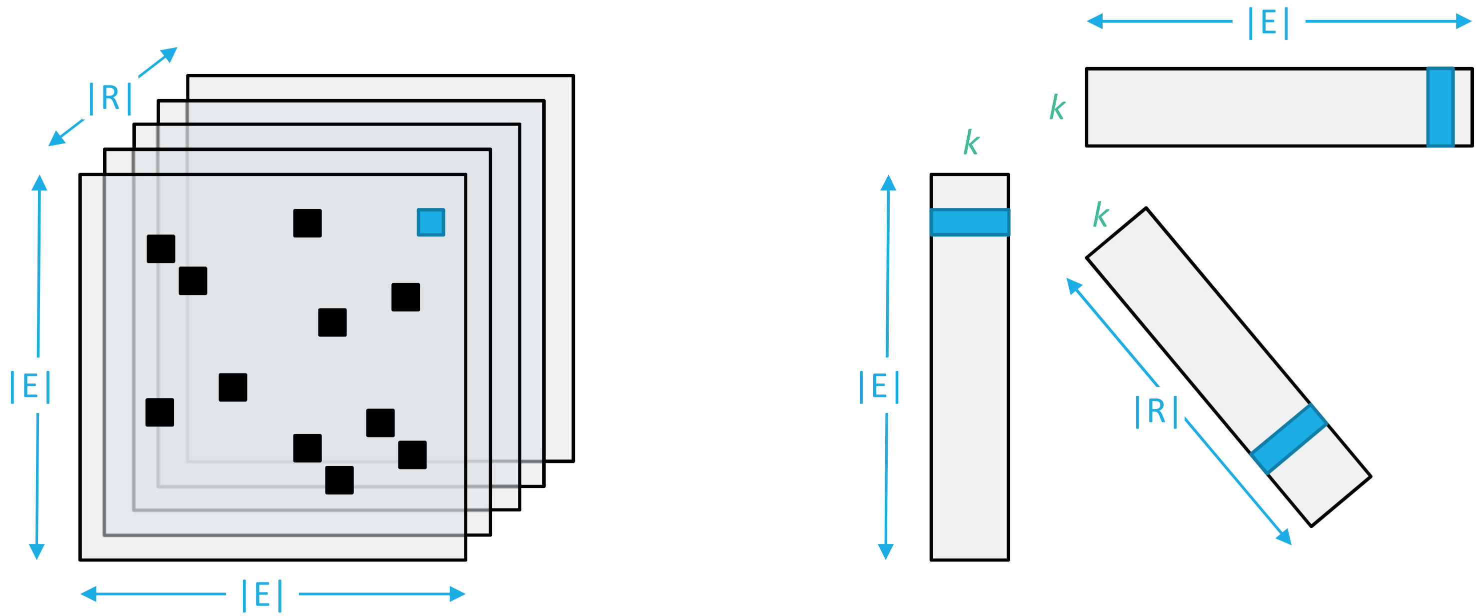


Tensor Formulation of KG



Does an unseen relation exist?

Factorize that Tensor



$$S(r(a,b)) = f(\mathbf{v}_r, \mathbf{v}_a, \mathbf{v}_b)$$

Many Different Factorizations

CANDECOMP/PARAFAC-Decomposition

$$S\left(r(a,b)\right) = \sum_k R_{r,k} \cdot e_{a,k} \cdot e_{b,k}$$

Tucker2 and RESCAL Decompositions

$$S\left(r(a,b)\right) = \left(\mathbf{R}_r \times \mathbf{e}_a\right) \times \mathbf{e}_b$$

Model E

$$S\left(r(a,b)\right) = \mathbf{R}_{r,1} \cdot \mathbf{e}_a + \mathbf{R}_{r,2} \cdot \mathbf{e}_b$$

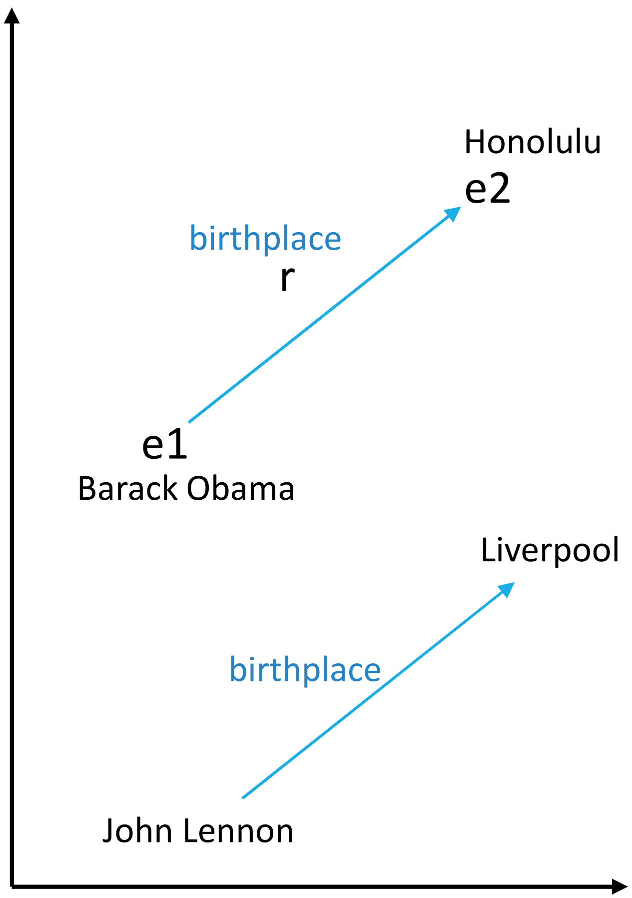
Holographic Embeddings

$$S(r(a,b)) = \mathbf{R}_r \times (\mathbf{e}_a \star \mathbf{e}_b)$$

Not tensor factorization (per se)

HOLE: Nickel et al, AAAI (2016), Model E: Riedel et al, NAACL (2013), RESCAL: Nickel et al, WWW (2012), CP: Harshman (1970), Tucker2: Tucker (1966)

Translation Embeddings



TransE

$$S\left(r(a,b)\right) = -\|\mathbf{e}_a + \mathbf{R}_r - \mathbf{e}_b\|_2^2$$

TransH

$$S\left(r(a,b)\right) = -\|\mathbf{e}_a^{\perp} + \mathbf{R}_r - \mathbf{e}_b^{\perp}\|_2^2$$

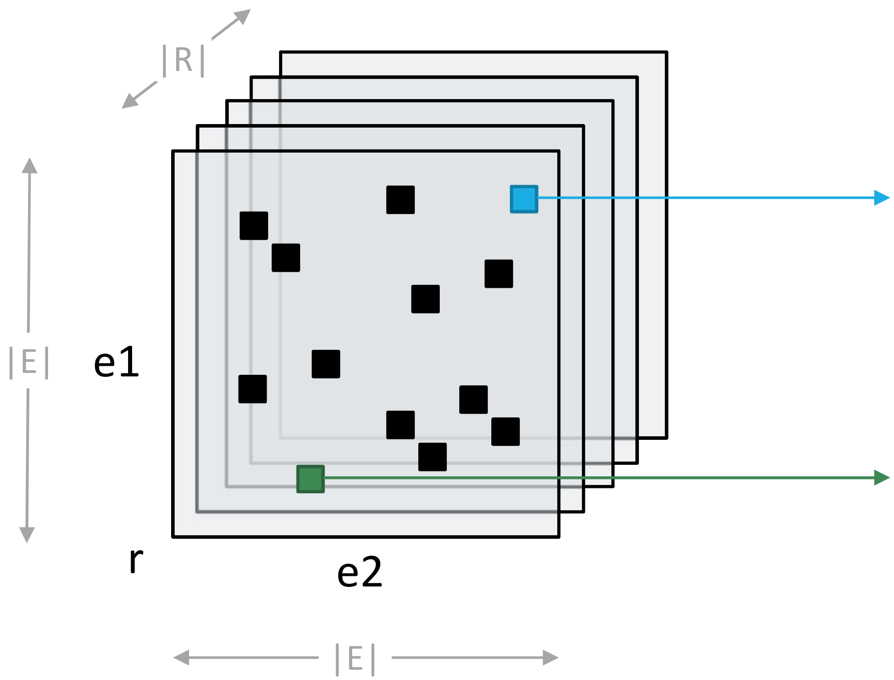
$$\mathbf{e}_a^{\perp} = \mathbf{e}_a - \mathbf{w}_r^T \mathbf{e}_a \mathbf{w}_r$$

TransR

$$S\left(r(a,b)\right) = -\|\mathbf{e}_a \mathbf{M}_r + \mathbf{R}_r - \mathbf{e}_b \mathbf{M}_r\|_2^2$$

TransE: Bordes et al. XXX (2011), TransH: Bordes et al. XXX (2011), TransR: Bordes et al. XXX (2011)

Parameter Estimation



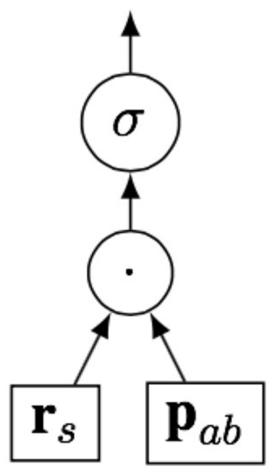
Observed cell: increase score

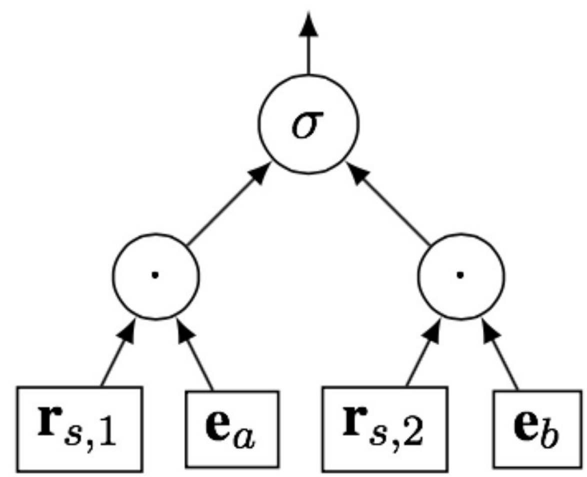
$$S\left(r(a,b)\right)$$

Unobserved cell: decrease score

$$S\left(r'(x,y)\right)$$

Matrix vs Tensor Factorization

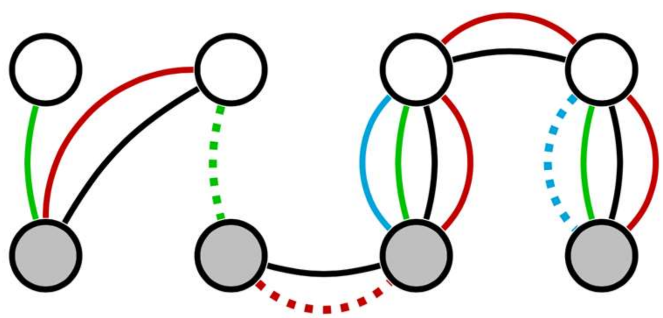




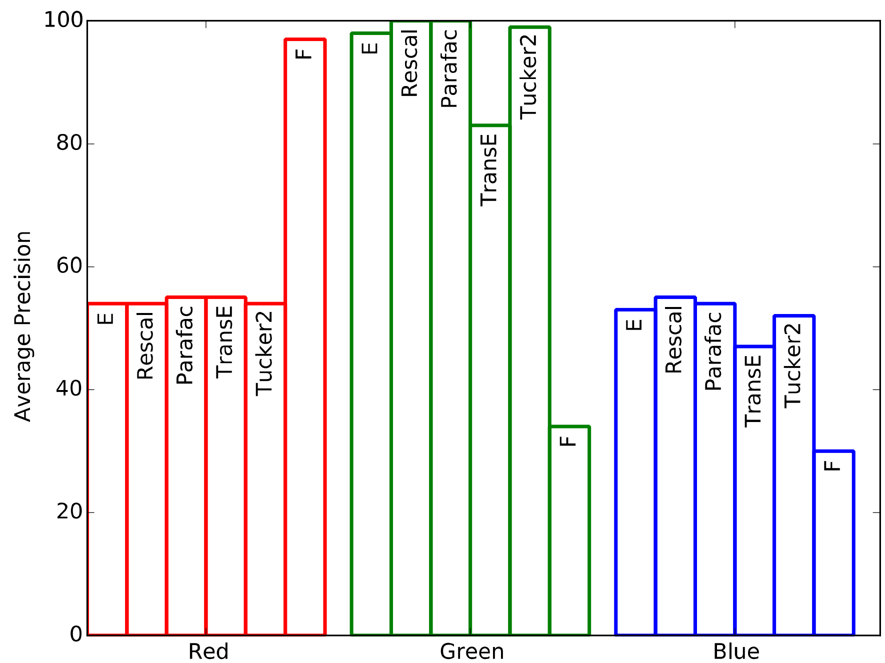
- Vectors for each entity pair
- Can only predict for entity pairs that appear in text together
- No sharing for same entity in different entity pairs

- Vectors for each entity
- Assume entity pairs are “low-rank”
  - But many relations are not!
  - Spouse: you can have only ~1
- Cannot learn pair specific information

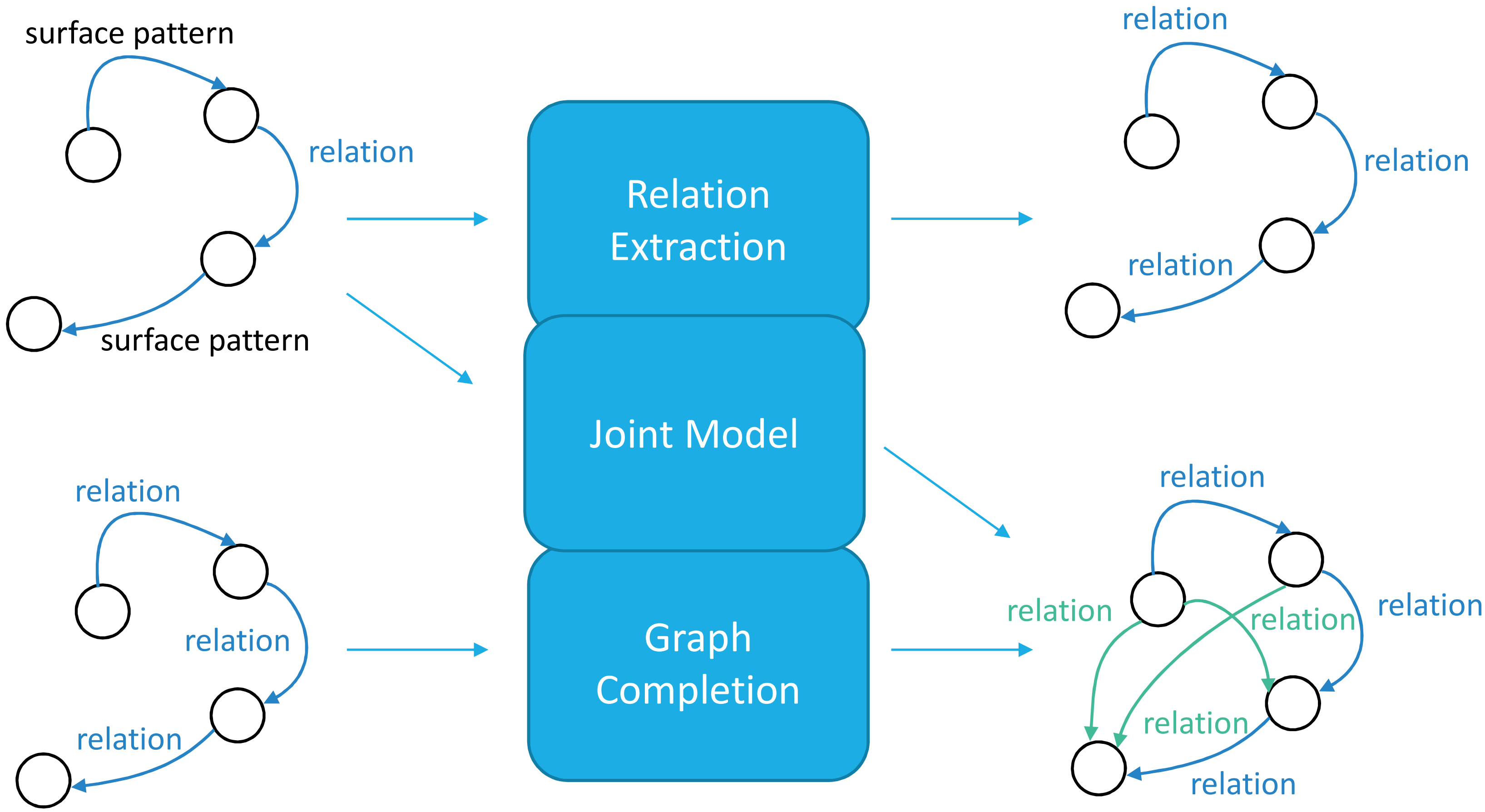
What they can, and can't, do..



- **Red**: deterministically implied by **Black**
  - needs *pair-specific* embedding
  - Only **F** is able to generalize
- **Green**: needs to estimate entity types
  - needs *entity-specific* embedding
  - Tensor factorization generalizes, **F** doesn't
- **Blue**: implied by **Red** and **Green**
  - Nothing works much better than random



Joint Extraction+Completion



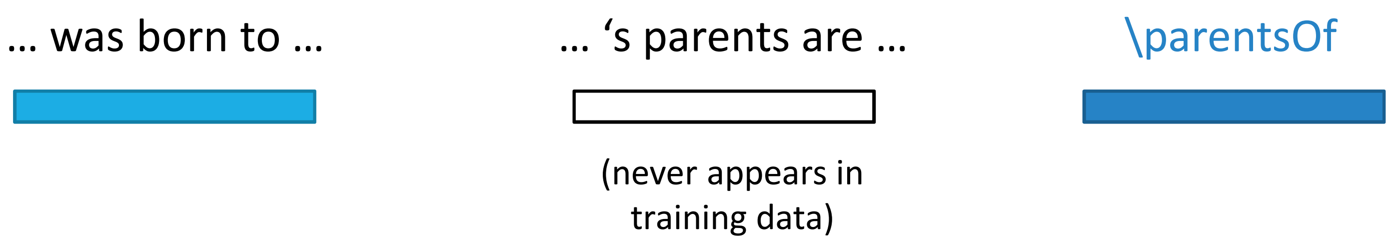
Compositional Neural Models

So far, we're learning vectors for each entity/surface pattern/relation..

But learning vectors independently ignores "composition"

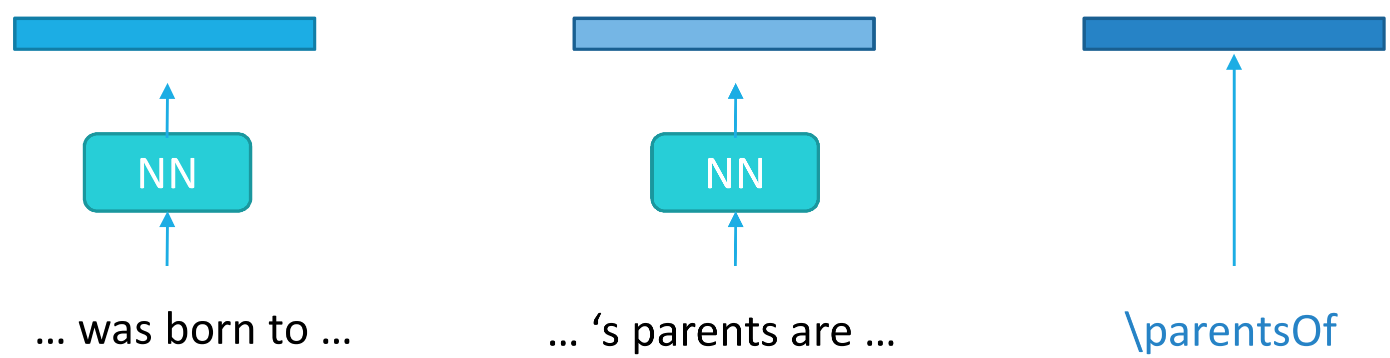
Composition in Surface Patterns		Composition in Relation Paths	
• Every surface pattern is not unique		• Every relation path is not unique	
• Synonymy:	A is B's spouse. A is married to B.	• Explicit:	A parent B, B parent C A grandparent C
• Inverse:	X is Y's parent. Y is one of X's children.	• Implicit:	X bornInCity Y, Y cityInState Z X "bornInState" Z
• Can the representation learn this?		• Can the representation capture this?	

Composing Dependency Paths

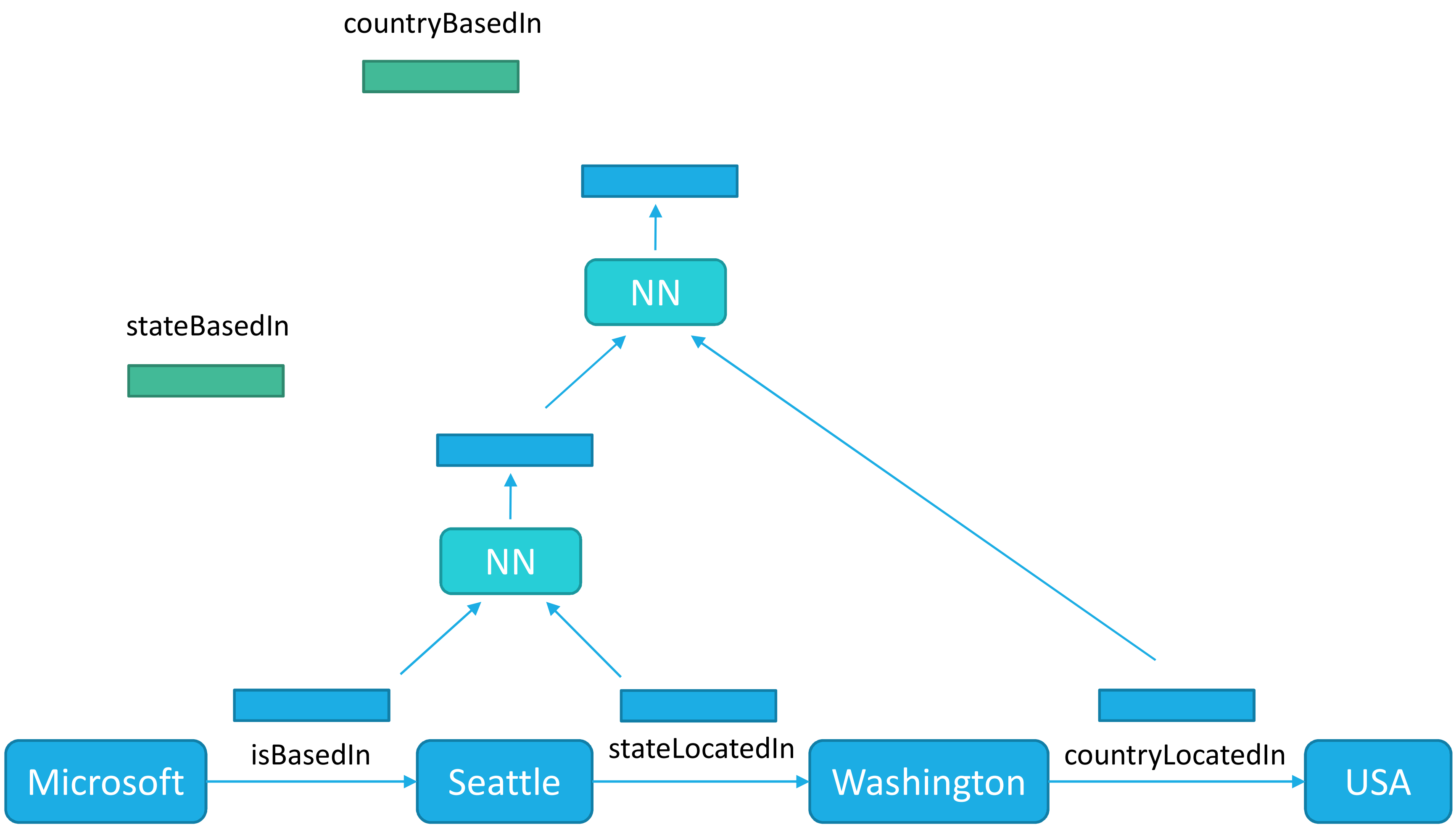


But we don’t need linked data to know they mean similar things…

Use neural networks to produce the embeddings from text!



Composing Relational Paths



Neelakantan et al (2015), http://www.aaai.org/ocs/index.php/SSS/SSS15/paper/viewFile/10254/10032

Lin et al, EMNLP (2015), https://arxiv.org/pdf/1506.00379.pdf

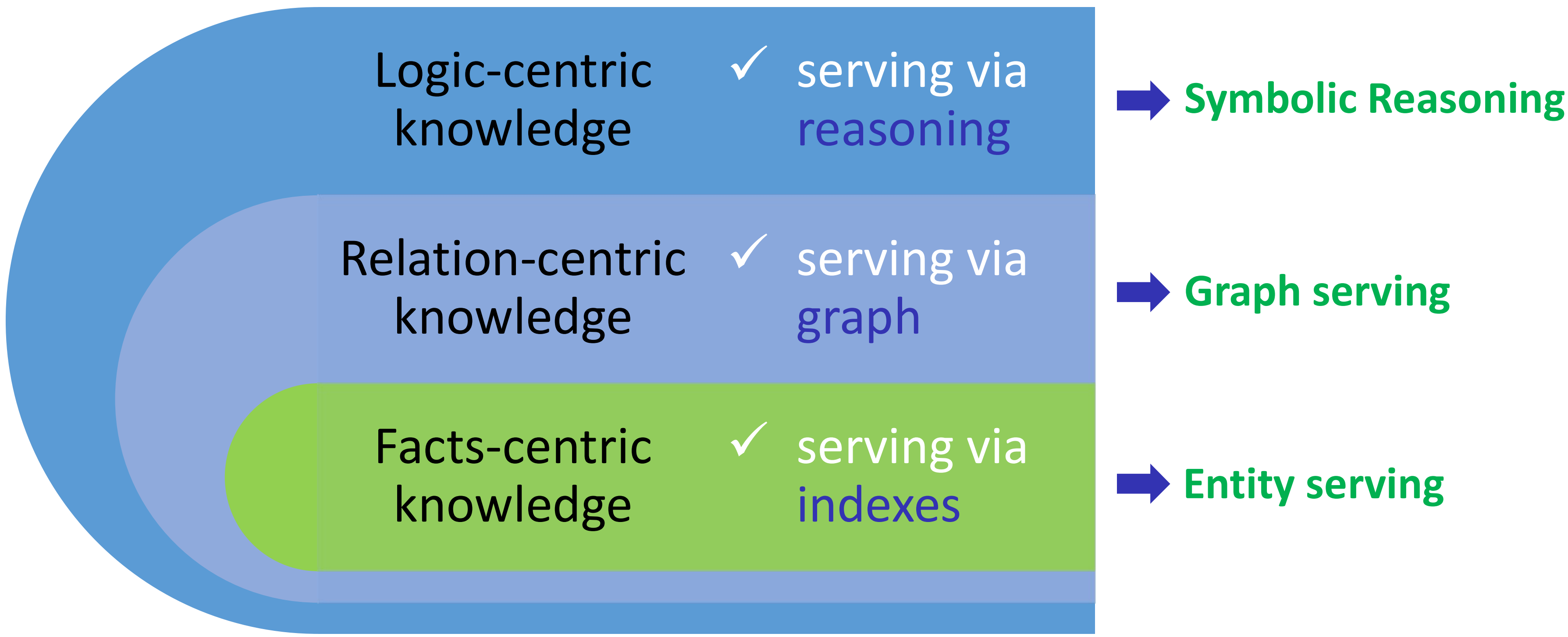
Review: Embedding Techniques

Two Related Tasks:
- Relation Extraction from Text
- Graph (or Link) Completion

Relation Extraction:
- Matrix Factorization Approaches

Graph Completion:
- Tensor Factorization Approaches

Compositional Neural Models
- Compose over dependency paths
- Compose over relation paths

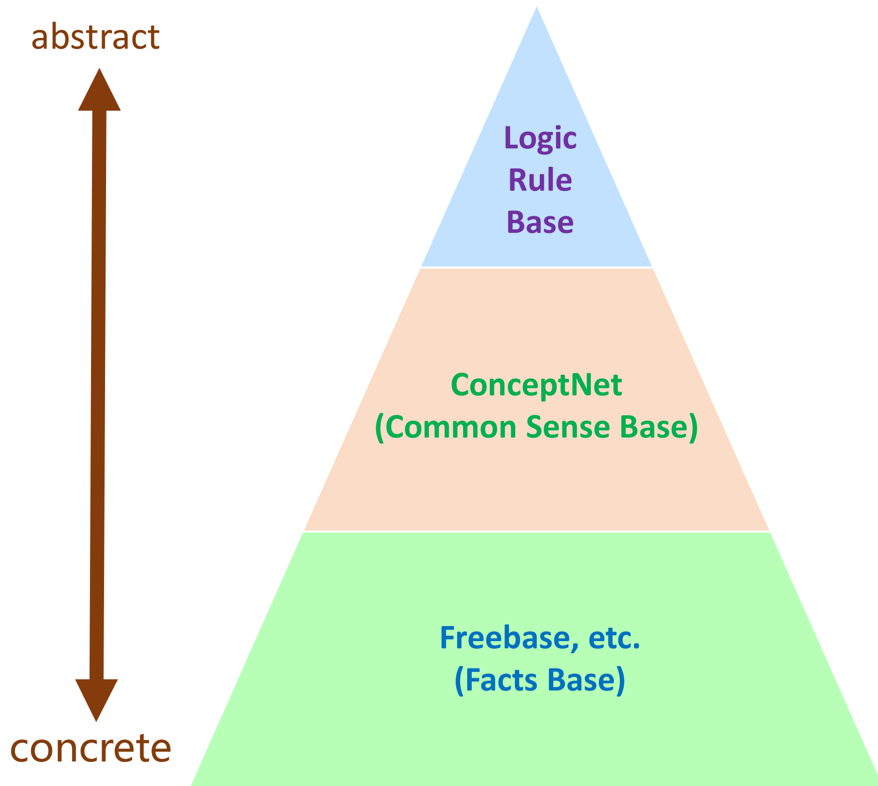


The evolution of knowledge representation

Why is a big knowledge graph not enough?

- Large knowledge graphs have billions of facts
- *However*, it doesn't provide much help in logic reasoning
    - The knowledge is not symbolized logic knowledge
    - Lack of reasoning rules allow machines to do reasoning automatically
    - More importantly, lack of common sense

The pyramid of knowledge



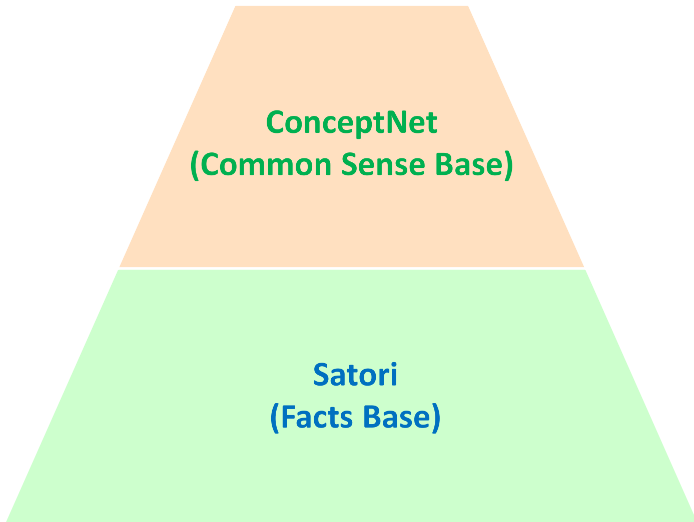
Knowledge in symbolic logic form

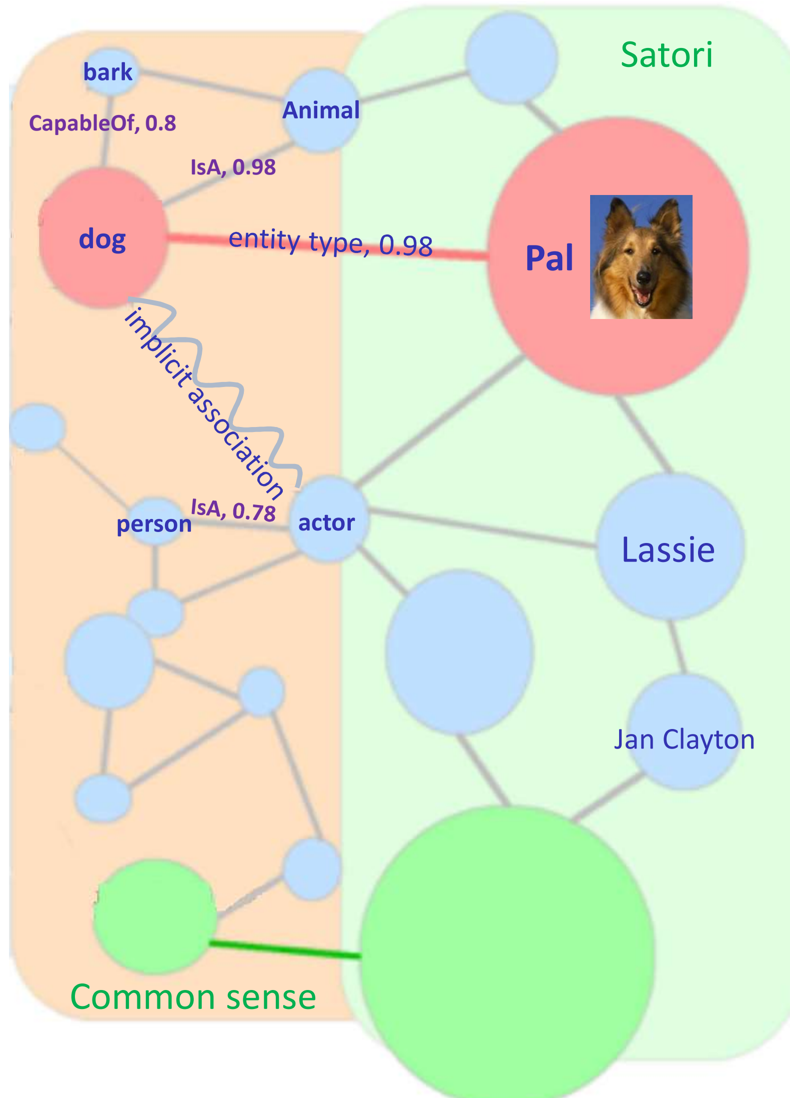
- **Symbols** are abstract identifiers can be manipulated in an algebra system
  - Variables
  - Functions

- Symbolic **expression** is a finite combination of symbols

- Symbolic **transformation**: a symbolic expression can be transformed into another symbolic expression according to the rules of a predefined reasoning algebra
  - An inference engine tries to derive answers for a logic question by performing logical deductions

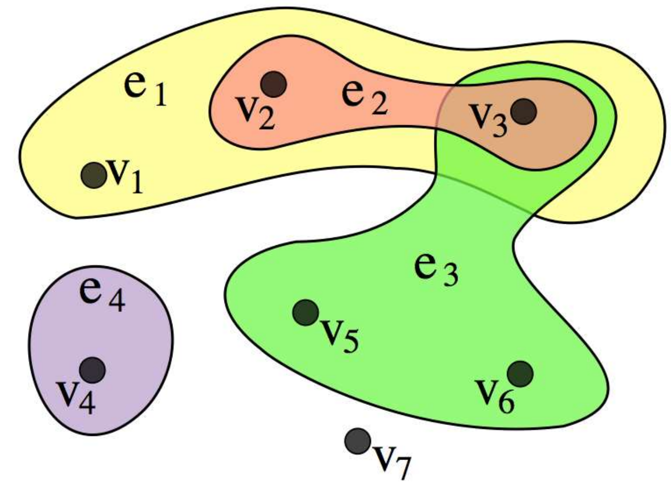
Represents Satori facts and common sense knowledge in RHHG



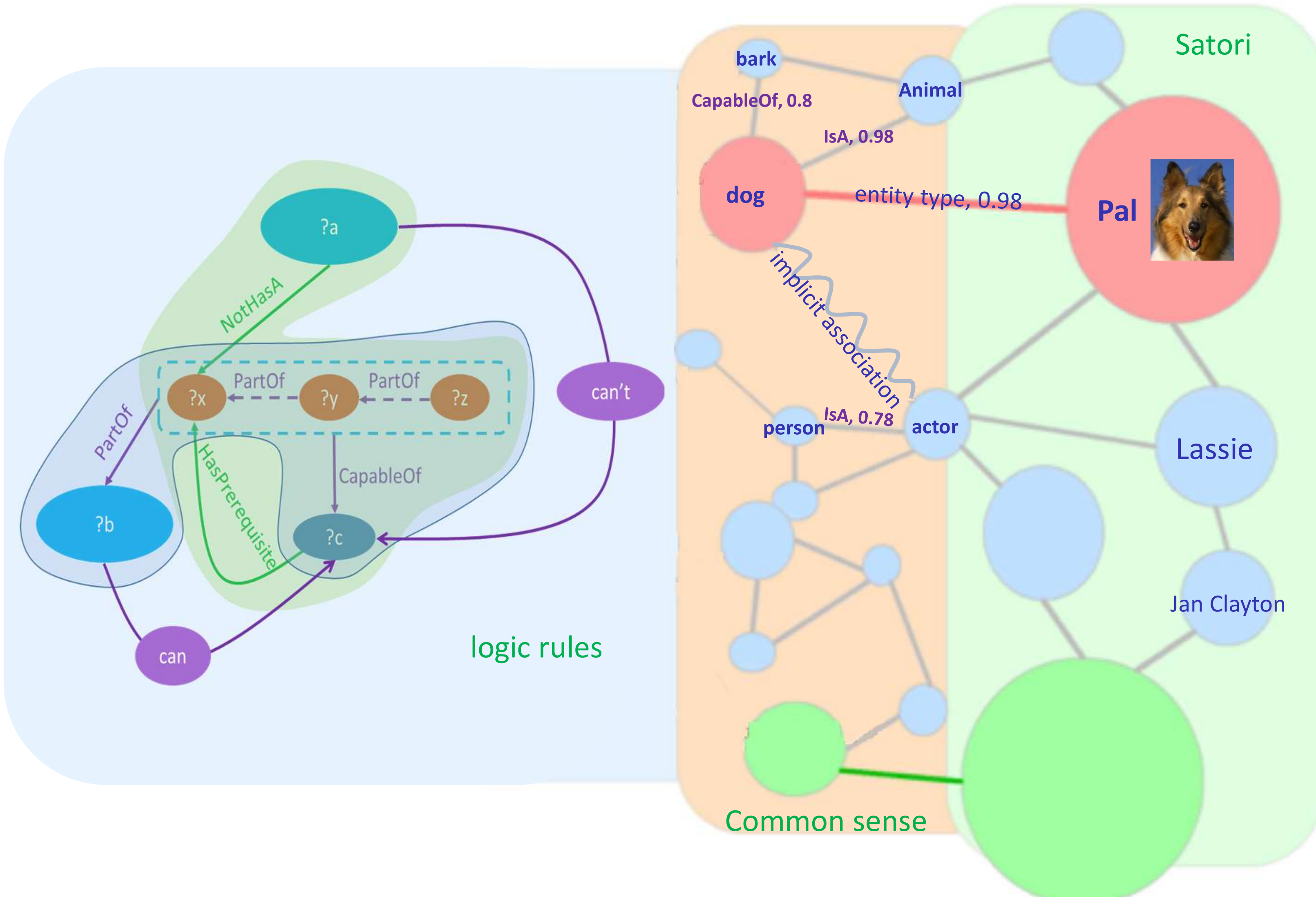


Functions and relations are just hyperedges!

- $f(x, y, z)$ is just a hyperedge f connecting three nodes x, y, z.
- A logical expression a AND b AND c can be written as $\mathrm{AND}(a, b, c)$.
- Symbolic transformation is just graph pattern matching and graph transformation!

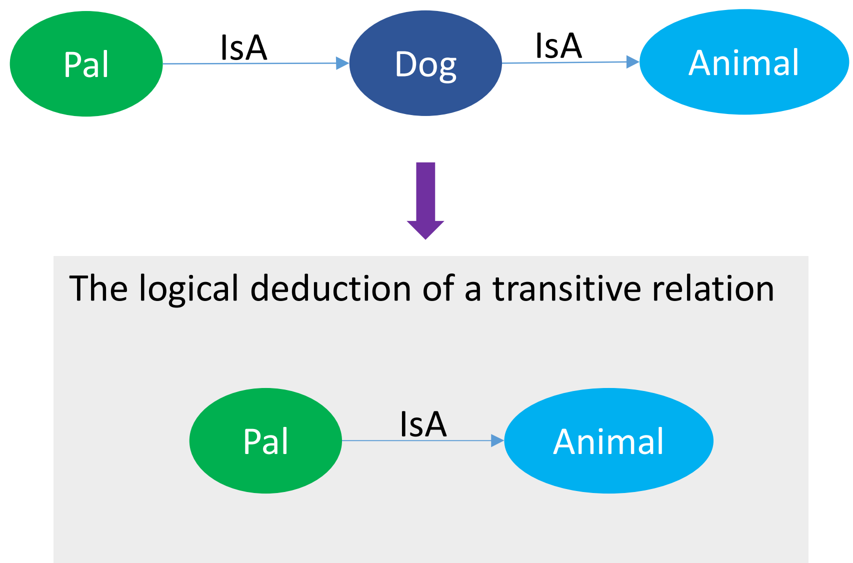


Hyperedges



?a
NotHasA
?x
PartOf
?y
PartOf
?z
PartOf
?b
HasPrerequisite
CapableOf
?c
can't
can
logic rules
bark
Animal
CapableOf, 0.8
IsA, 0.98
dog
entity type, 0.98
Pal
implicit association
person
IsA, 0.78
actor
Lassie
Jan Clayton
Satori
Common sense

Use graph transformation to do logic deduction

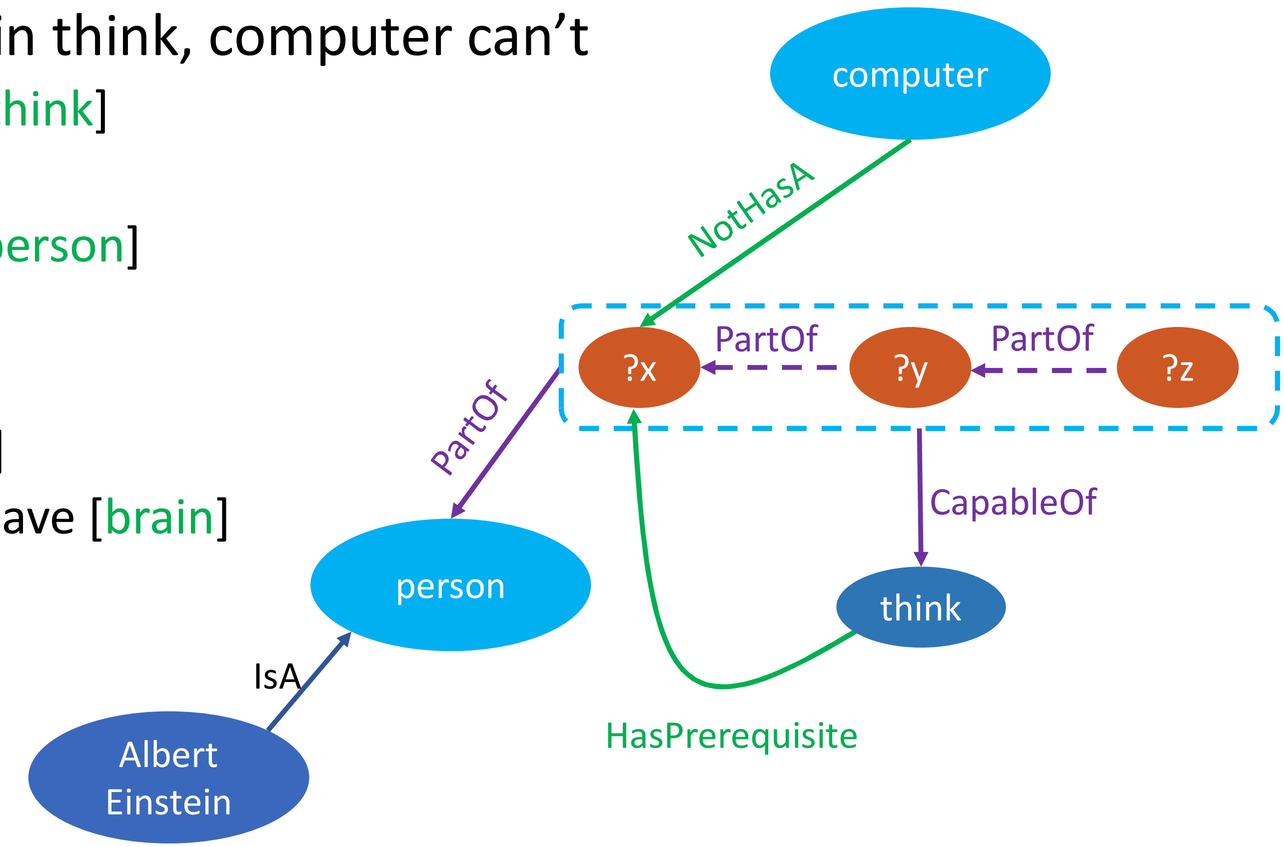


Graph transformation: whenever we see a graph G_a with a certain pattern p, replace it with a graph G_b.

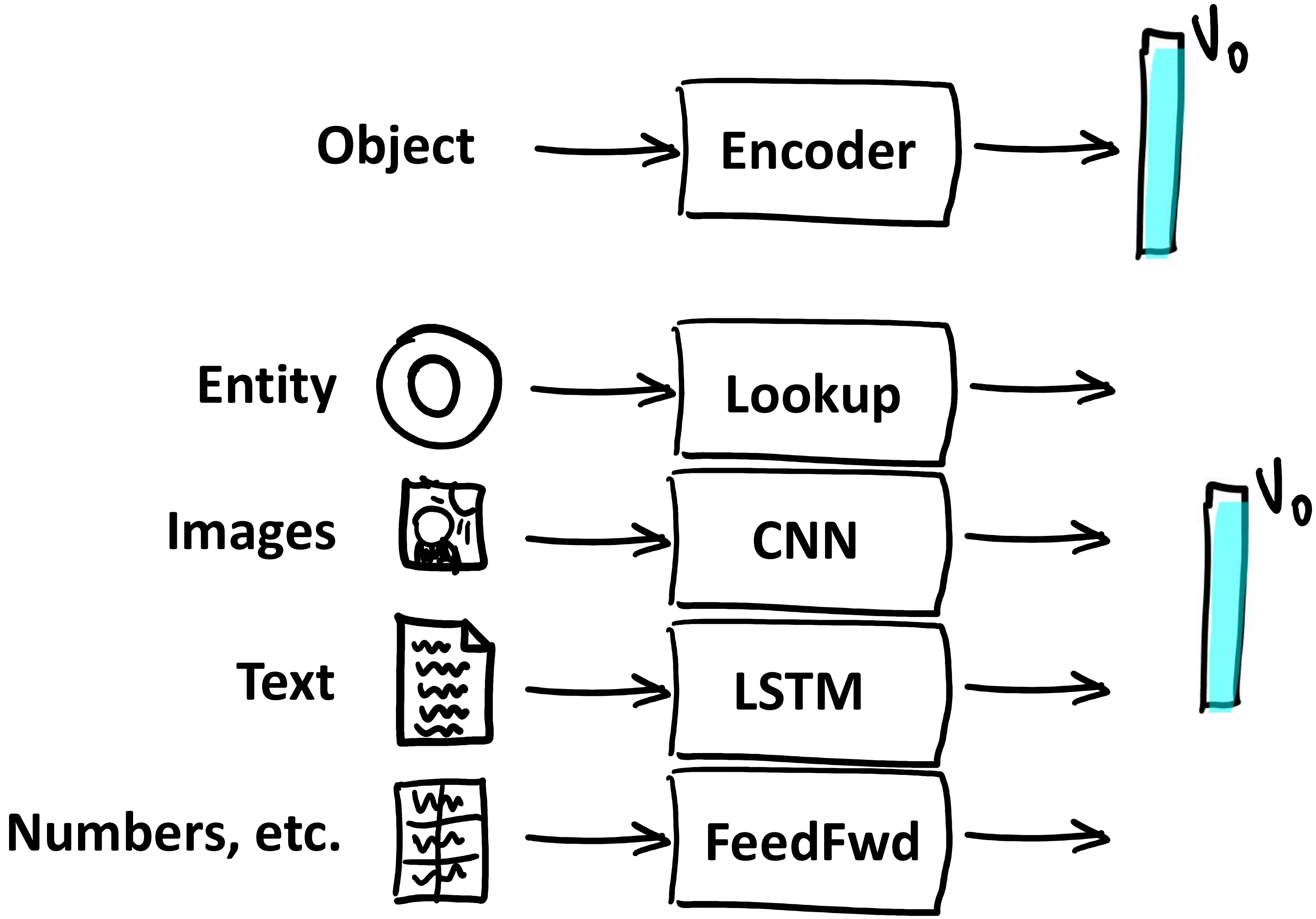
Our “shallow” yet reasonable answer

- Why can Albert Einstein think, computer can’t
  - [brain] is Capable Of [think]
  - [person] have [brain]
  - [Albert Einstein] is a [person]

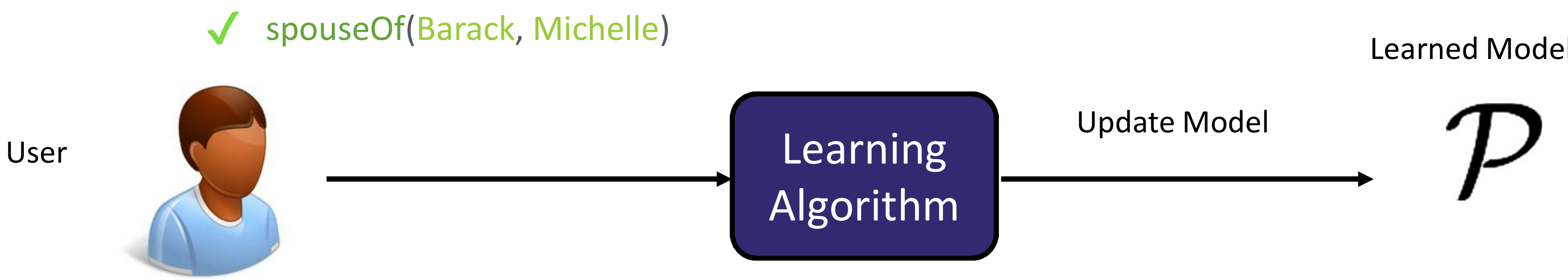
  - [think] requires [brain]
  - [computer] does not have [brain]



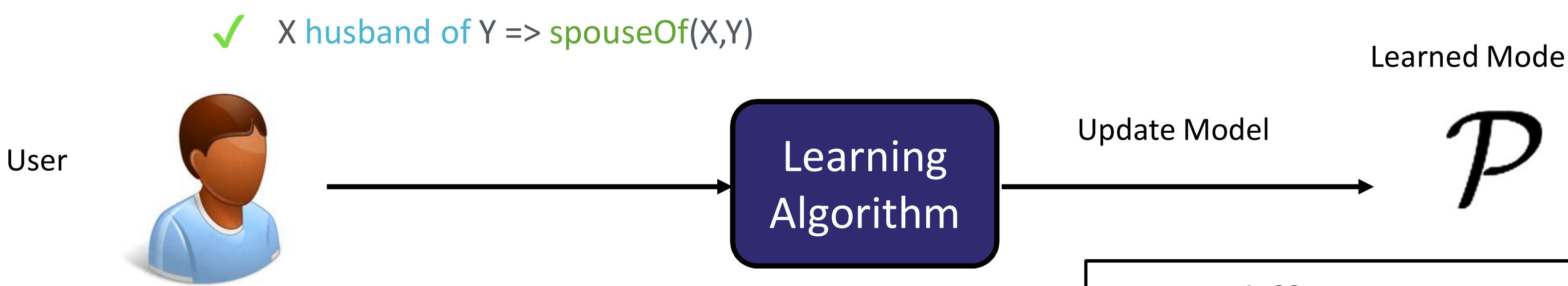
Multimodal KB Embeddings



Knowledge as Supervision



Problem 1: Each annotation takes time
Problem 2: Each annotation is a drop in the ocean



Many different options
- Generalized Expectation
- Posterior Regularization
- Labeling functions in SNORKEL

(2) Future research directions: Online KG Construction

- One shot KG construction → Online KG construction
  - Consume online stream of data
  - Temporal scoping of facts
  - Discovering new concepts automatically
  - Self-correcting systems

(2) Future research directions:
Online KG Construction

- **Continuously learning and self-correcting systems**
  - *[Selecting Actions for Resource-bounded Information Extraction using Reinforcement Learning, Kanani and McCallum, WSDM 2012]*
    - Presented a reinforcement learning framework for budget constrained information extraction
  - *[Never-Ending Learning, Mitchell et al. AAAI 2015]*
    - Tom Mitchell says “Self reflection and an explicit agenda of learning subgoals” is an important direction of future research for continuously learning systems.