



AAAI 2018 Tutorial

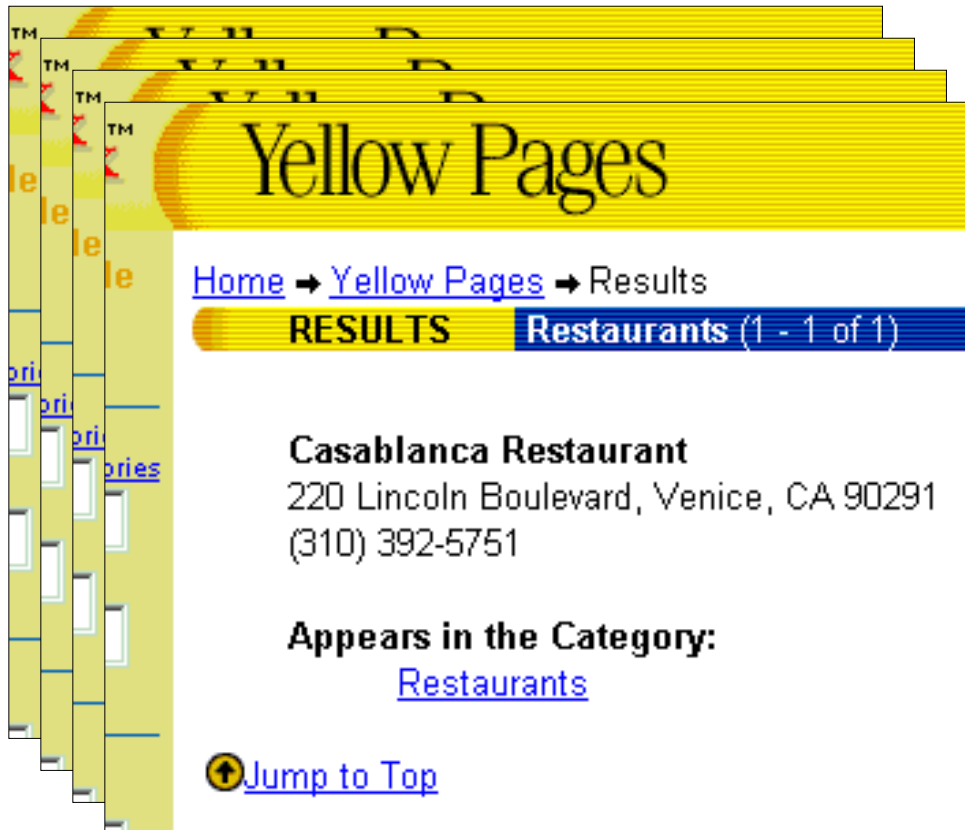
Building Knowledge Graphs

Craig Knoblock

University of Southern California

Wrappers for Web Data Extraction

Extracting Data from Semi-structured Sources



Yellow Pages

[Home](#) → [Yellow Pages](#) → Results

RESULTS Restaurants (1 - 1 of 1)

Casablanca Restaurant
220 Lincoln Boulevard, Venice, CA 90291
(310) 392-5751

Appears in the Category:
[Restaurants](#)

[↑ Jump to Top](#)

NAME	Casablanca Restaurant
STREET	220 Lincoln Boulevard
CITY	Venice
PHONE	(310) 392-5751

Approaches to Wrapper Construction

- Manual Wrapper Construction
- Learning Wrappers from Labelled Examples
- Grammar Induction for Automatic Wrapper Construction

Grammar Induction Approach

- Pages automatically generated by scripts that encode results of db query into HTML
 - Script = grammar
- Given a set of pages generated by the same script
 - Learn the grammar of the pages
 - Wrapper induction step
 - Use the grammar to parse the pages
 - Data extraction step

RoadRunner

[Crescenzi, Mecca, & Merialdo]

- Automatically generates a wrapper from large web pages
 - Pages of the same *class*
 - No dynamic content from javascript, ajax, etc
- Infers source schema
 - Supports nested structures and lists
 - Extracts data from pages
- Efficient approach to large, complex pages with regular structure

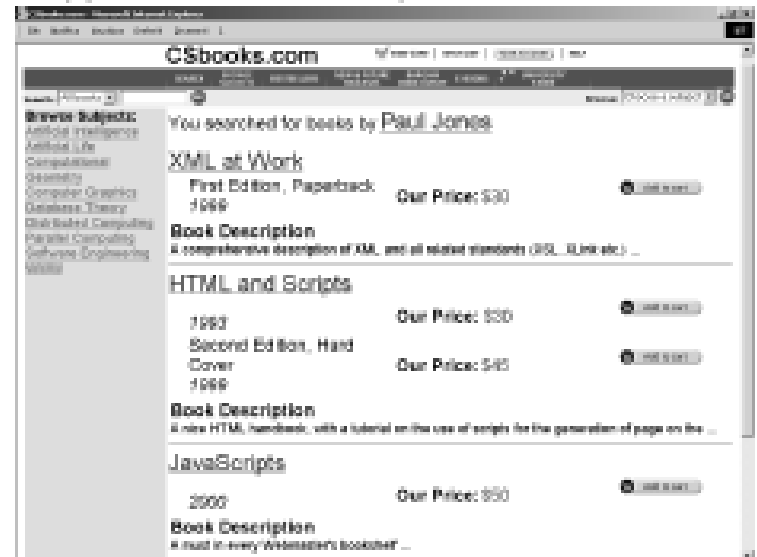
Example Pages

- Compares two pages at a time to find similarities and differences
- Infers nested structure (schema) of page
- Extracts fields

<http://www.csbooks.com/author?John+Smith>



<http://www.csbooks.com/author?Paul+Jones>



Extracted Result

Microsoft Access - Microsoft Access Database

File Edit View Database Window Help

Total number of SCHEMAS found: 1

Schema Number 1: A | B | C | D | E | F | Total Time: 0" 150 ms

sample1/acc1

A					F
John Smith	Database Primer	C	D	E	This book introduces the reader to the theory and technology... (TRUNCATED)
		First Edition, Paperback	1998	\$20	
		Second Edition, Hard Cover	2000	\$30	
	Computer Systems	First Edition, Paperback	1996	\$40	An undergraduate level introduction to computer... (TRUNCATED)

sample2/acc1

A					F
Paul Jones	XML at Work	C	D	E	A comprehensive description of XML, and all related standards... (TRUNCATED)
		First Edition, Paperback	1999	\$30	
	HTML and Scripts	null	1993	\$30	A useful HTML handbook, with a good tutorial on the use of sc... (TRUNCATED)
		Second Edition, Hard Cover	1999	\$45	
	JavaScripts	null	2000	\$50	A must in every Webmaster's bookshelf ...

Union-Free Regular Expression (UFRE)

- Web page structure can be represented as *Union-Free Regular Expression* (UFRE)
 - UFRE is Regular Expressions without *disjunctions*
 - If a and b are UFRE, then the following are also UFREs
 - $a.b$
 - $(a)^+$
 - $(a)^?$

Union-Free Regular Expression (UFRE)

- Web page structure can be represented as *Union-Free Regular Expression* (UFRE)
 - UFRE is Regular Expressions without *disjunctions*
 - If a and b are UFRE, then the following are also UFREs
 - $a.b \rightarrow$ string fields
 - $(a)^+ \rightarrow$ lists (possibly nested)
 - $(a)^? \rightarrow$ optional fields
 - Strong assumption that usually holds

Approach

- Given a set of example pages
- Generate the *Union-Free Regular Expression* which contains example pages
- Find the least upper bounds on the RE lattice to generate a wrapper in *linear time*
- Reduces to finding the least upper bound on two UFREs

Matching/Mismatches

Given a set of pages of the same type

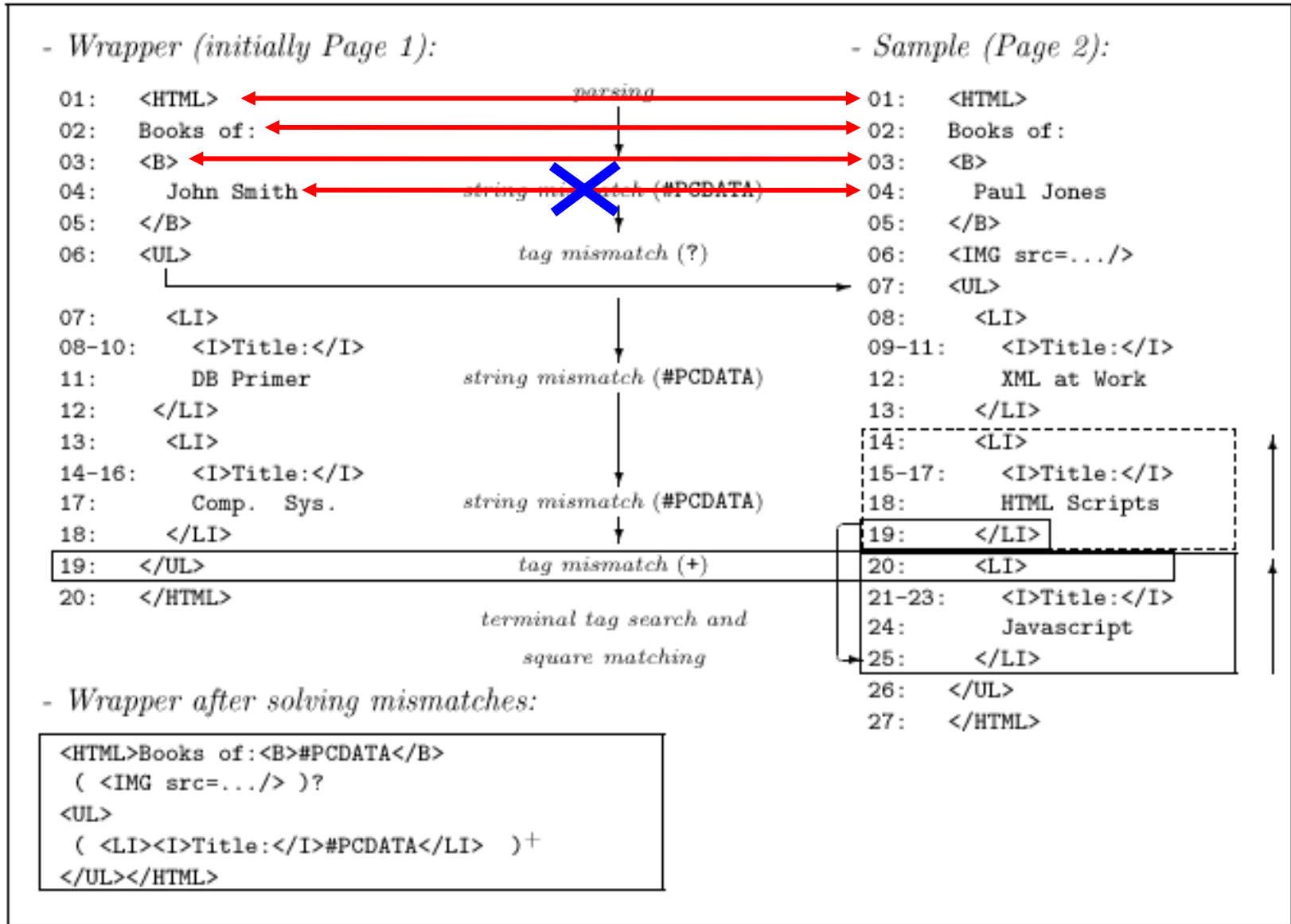
- Take the first page to be the *wrapper* (UFRE)
- Match each successive sample page against the wrapper
- *Mismatches* result in generalizations of wrapper
 - String mismatches
 - Tag mismatches

Matching/Mismatches

Given a set of pages of the same type

- Take the first page to be the *wrapper* (UFRE)
- Match each successive sample page against the wrapper
- *Mismatches* result in generalizations of wrapper
 - String mismatches
 - Discover fields
 - Tag mismatches
 - Discover optional fields
 - Discover iterators

Example Matching



String Mismatches: Discovering Fields

- String mismatches are used to discover fields of the document
- Wrapper is generalized by replacing “John Smith” with #PCDATA

<HTML>Books of: John Smith

→ <HTML> Books of: #PCDATA

Example Matching

- Wrapper (initially Page 1):

```

01: <HTML>
02: Books of:
03: <B>
04:   John Smith
05: </B>
06: <UL>
07:   <LI>
08-10:   <I>Title:</I>
11:     DB Primer
12:   </LI>
13:   <LI>
14-16:   <I>Title:</I>
17:     Comp. Sys.
18:   </LI>
19: </UL>
20: </HTML>
    
```

- Sample (Page 2):

```

01: <HTML>
02: Books of:
03: <B>
04:   Paul Jones
05: </B>
06: <IMG src=.../>
07: <UL>
08:   <LI>
09-11:   <I>Title:</I>
12:     XML at Work
13:   </LI>
14:   <LI>
15-17:   <I>Title:</I>
18:     HTML Scripts
19:   </LI>
20:   <LI>
21-23:   <I>Title:</I>
24:     Javascript
25:   </LI>
26: </UL>
27: </HTML>
    
```

parsing

string mismatch (#PCDATA)

tag mismatch (?)

string mismatch (#PCDATA)

string mismatch (#PCDATA)

tag mismatch (+)

terminal tag search and
square matching



- Wrapper after solving mismatches:

```

<HTML>Books of:<B>#PCDATA</B>
 ( <IMG src=.../> )?
<UL>
 ( <LI><I>Title:</I>#PCDATA</LI> )+
</UL></HTML>
    
```


Tag Mismatches: Discovering Optionals

- First check to see if mismatch is caused by an iterator (described next)
- If not, could be an optional field in wrapper *or* sample
- Cross search used to determine possible optionals
- Image field determined to be optional:
 - (``)?

Example Matching

- Wrapper (initially Page 1):

```

01: <HTML>
02: Books of:
03: <B>
04:   John Smith
05: </B>
06: <UL>
07:   <LI>
08-10:   <I>Title:</I>
11:     DB Primer
12:   </LI>
13:   <LI>
14-16:   <I>Title:</I>
17:     Comp. Sys.
18:   </LI>
19: </UL>
20: </HTML>
    
```

- Sample (Page 2):

```

01: <HTML>
02: Books of:
03: <B>
04:   Paul Jones
05: </B>
06: <IMG src=.../>
07: <UL>
08:   <LI>
09-11:   <I>Title:</I>
12:     XML at Work
13:   </LI>
14:   <LI>
15-17:   <I>Title:</I>
18:     HTML Scripts
19:   </LI>
20:   <LI>
21-23:   <I>Title:</I>
24:     Javascript
25:   </LI>
26: </UL>
27: </HTML>
    
```

parsing
 ↓
 string mismatch (#PCDATA)
 ↓
 tag mismatch (?)

String Mismatch

String Mismatch

tag mismatch (+)

terminal tag search and
 square matching

- Wrapper after solving mismatches:

```

<HTML>Books of:<B>#PCDATA</B>
 ( <IMG src=.../> )?
<UL>
 ( <LI><I>Title:</I>#PCDATA</LI> )+
</UL></HTML>
    
```

Tag Mismatches: Discovering Iterators

- Assume mismatch is caused by repeated elements in a list
 - End of the list corresponds to last matching token: ``
 - Beginning of list corresponds to one of the mismatched tokens: `` or ``
 - These create possible “squares”
- Match possible squares against earlier squares
- Generalize the wrapper by finding all contiguous repeated occurrences:
 - `(<I>Title:</I>#PCDATA)+`

Example Matching

- Wrapper (initially Page 1):

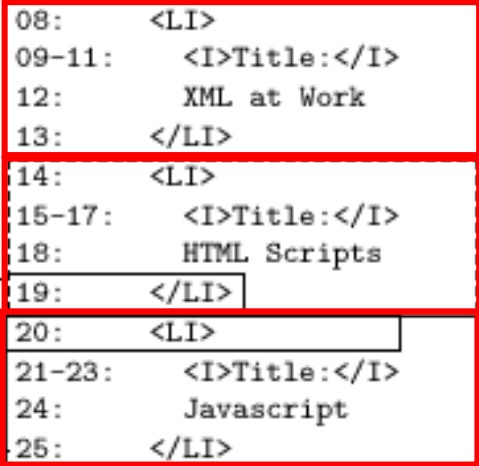
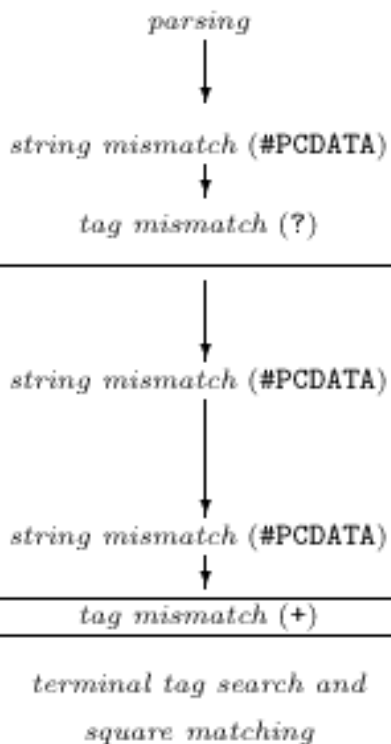
```

01: <HTML>
02: Books of:
03: <B>
04:   John Smith
05: </B>
06: <UL>
07:   <LI>
08-10:   <I>Title:</I>
11:     DB Primer
12:   </LI>
13:   <LI>
14-16:   <I>Title:</I>
17:     Comp. Sys.
18:   </LI>
19: </UL>
20: </HTML>
    
```

- Sample (Page 2):

```

01: <HTML>
02: Books of:
03: <B>
04:   Paul Jones
05: </B>
06: <IMG src=.../>
07: <UL>
08:   <LI>
09-11:   <I>Title:</I>
12:     XML at Work
13:   </LI>
14:   <LI>
15-17:   <I>Title:</I>
18:     HTML Scripts
19:   </LI>
20:   <LI>
21-23:   <I>Title:</I>
24:     Javascript
25:   </LI>
26: </UL>
27: </HTML>
    
```



- Wrapper after solving mismatches:

```

<HTML>Books of:<B>#PCDATA</B>
 ( <IMG src=.../> )?
<UL>
 ( <LI><I>Title:</I>#PCDATA</LI> )+
</UL></HTML>
    
```

Internal Mismatches

- Generate *internal mismatch* while trying to match square against earlier squares on *the same page*
- Solving internal mismatches yield further refinements in the wrapper
 - List of book editions
 - `<I>Special!</I>`

Recursive Example

- Wrapper (initially Page 1):

```

01-05: <HTML>Books of:<B>John Smith</B>
06:   <UL>
07:     <LI>
08:       Computer Systems
09:       <P>
10:         <B>
11:           1st Ed., 1995
12:         </B>
13:       </P>
14:     </LI>
15:     <LI>
16:       Database Primer
17:       <P>
18:         <B>
19:           1st Ed., 1998
20-22:       <I>Special!</I>
23:         </B>
24:         <B>
25:           2nd Ed., 2000
26:         </B>
27:       </P>
28:     </LI>
29-30: </UL></HTML>
        
```

internal mismatch →

- Wrapper after solving mismatches:

```

<HTML>Books of:<B>#PCDATA</B>
<UL>(<LI>#PCDATA<P>
  (<B>#PCDATA
    (<I>Special!</I>)?
    </B>)+ </P></LI>)+
</UL></HTML>
        
```

- Sample (Page 2):

```

01-05: <HTML>Books of:<B>Paul Jones</B>
06:   <UL>
07:     <LI>
08:       XML at Work
09:       <P>
10:         <B>
11:           1st Ed., 1999
12:         </B>
13:       </P>
14:     </LI>
15:   </UL>
16: </HTML>
        
```

external mismatch

```

28: </LI>
27: </P>
26: </B>
25:   2nd Ed., 2000
24: <B>
23: </B>
        
```

```

14: </LI>
13: </P>
12: </B>
11:   1st Ed., 1995
10: <B>
09: <P>
08:   Computer Systems:
07: <LI>
        
```

Discussion

- Assumptions:
 - Pages are well-structured
 - Structure can be modeled by UFRE (no disjunctions)
- Search space for explaining mismatches is huge
 - Uses a number of heuristics to prune space
 - Limited backtracking
 - Limit on number of choices to explore
 - Patterns cannot be delimited by optionals
 - Will result in pruning possible wrappers

Limitations

- Learnable grammars
 - Union-Free Regular Expressions (RoadRunner)
 - Variety of schema structure: tuples (with optional attributes) and lists of (nested) tuples
 - Does not efficiently handle disjunctions – pages with alternate presentations of the same attribute
 - Context-free Grammars
 - Limited learning ability
- User needs to provide a set of pages of the same type

Inferlink Web Extraction Software

Extraction

FOR SALE: STOEGER M3500

post id: 4700468

share: [f](#) [✉](#) [t](#) [p](#)

Price: \$ 500
Seller: Private Party
Account: Registered on 5/9/2013
[Listings by this user](#)

Listed On: Thursday, September 17, 2015
Listed In: Shotguns
Location: Keenesburg, Denver, Colorado - [Map](#)
Shipping: No

Manufacturer: Stoeger
Caliber: 12 Gauge
Action: Semi-automatic
Firearm Type: Shotgun

[Flag](#) | [Edit](#) | [Favorite](#)

[Contact Seller](#)

I have a Stoeger m3500. It is a year old. It has 200 rounds through it from clay shooting. Its in perfect condition. If you have any questions email or text me. 9703427061. I'm asking 500

[Contact Seller](#)



Structured Extraction

FOR SALE: STOEGER M3500

post id: 4700468

share: [f](#) [✉](#) [t](#) [p](#)

Price:

\$ 500

Seller:

Private Party

Account:

Registered on 5/9/2013

[Listings by this user](#)

Listed On:

Thursday, September 17, 2015

Listed In:

Shotguns

Location:

Keenesburg, Denver, Colorado [Map](#)

Shipping:

No

Manufacturer:

Stoeger

Caliber:

12 Gauge

Action:

Semi-automatic

Firearm Type:

Shotgun

[Flag](#) | [Edit](#) | [Favorite](#)

Contact Seller

I have a Stoeger m3500. It is a year old. It has 200 rounds through it from clay shooting. Its in perfect condition. If you have any questions email or text me. 9703427061. I'm asking 500

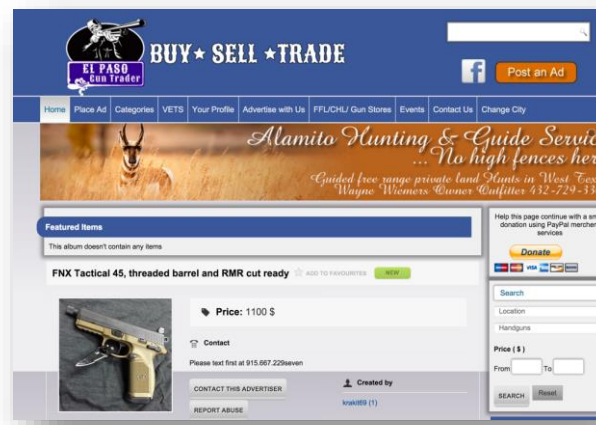
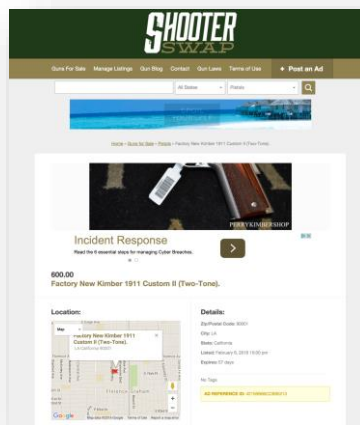
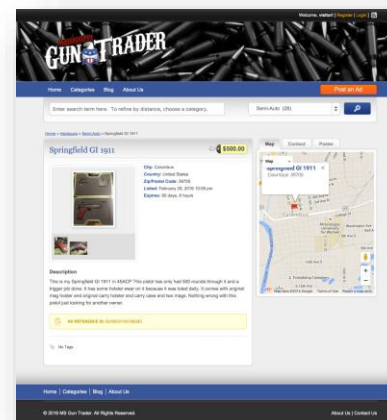
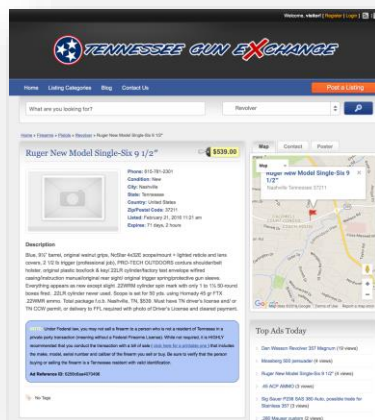
Contact Seller



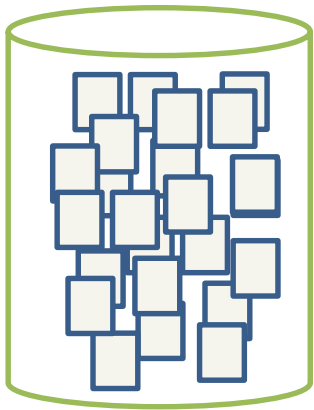
Automated Extraction

[Minton et al., Inferlink]

- Title
- Description
- Seller
- Post Date
- Expiry Date
- Price
- Location
- Category
- Member Since
- Num Views
- Post ID

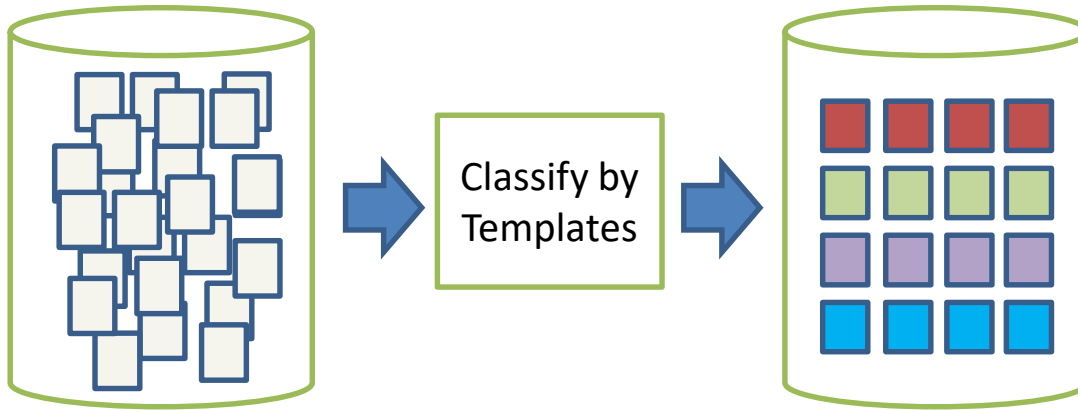


Automated Extraction



Input: A Pile of Pages

Automated Extraction



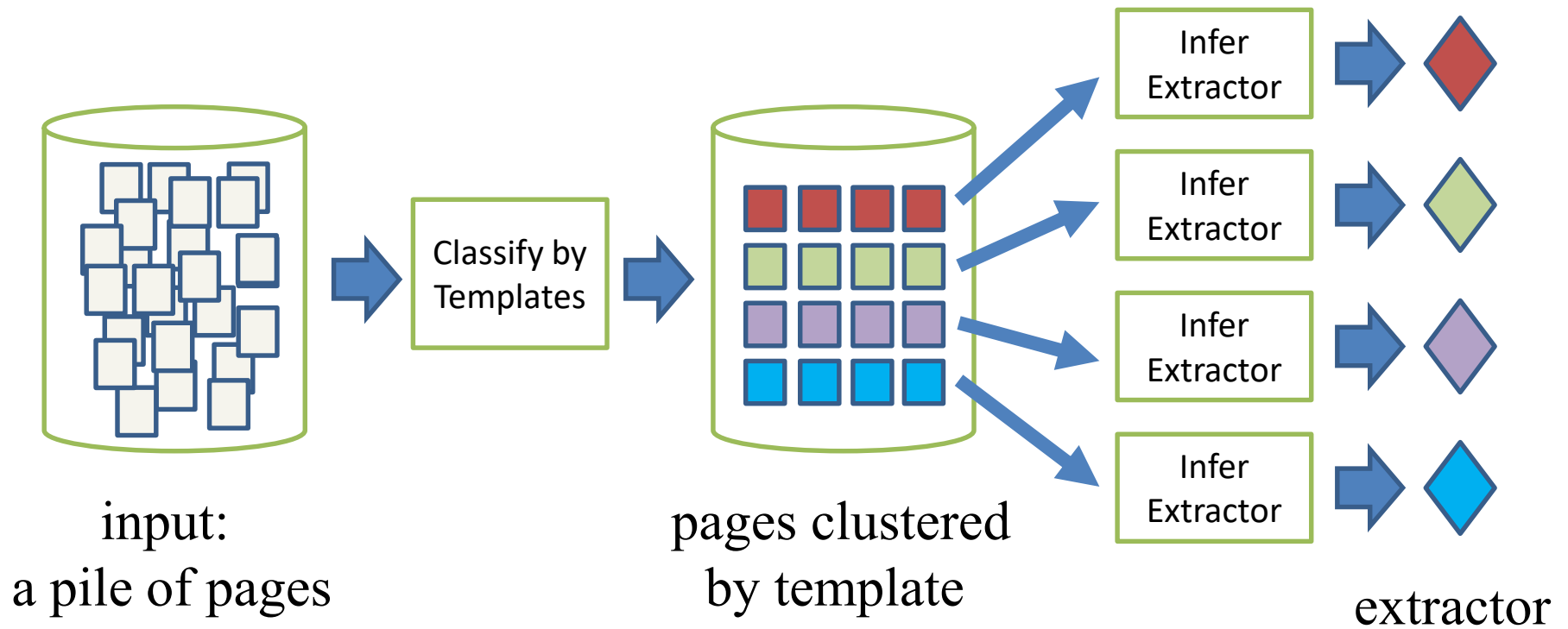
input:
a pile of pages

pages clustered
by template

Clustering

- Cluster
 - Based on the visible text
 - Page is broken into chunks
 - These are continuous blocks of text
 - Search for common visible chunks
 - Remove chunks that occur in all pages
 - Remove those that occur in fewer than 10 pages
 - Greedy algorithm to cluster the pages based on the remaining chunks
 - Sort by the size of the clusters created by each chunk

Automated Extraction



Extractor Learning

- Input: cluster
- Select 5 random pages to build a template
 - Tokenize on space & punctuation
 - Start with n-grams of tuples of size n
 - Find those n-grams that occur on all pages
 - Keep only those n-grams that occur exactly once per page
 - Decompose pages based on these n-grams
 - Run algorithm recursively on decomposed page
 - Repeat above for size $n-1$ down to $n=2$
 - Construct template based on the decomposition

Unsupervised Extraction Tool

InferLink {Landmark}

tennesseegunexchange

Markup

Extraction

Downloads



+ Add Page	page_2.html	page_3.html	page_4.html	page_5.html
0036	Winchester model 140	Beretta hand gun for sale	1956 Mossin nagant	Ruger New Model Single-Six 9 1/2″
Code1354	37122	37201	37122	37211
Description1392	Winchester model 140 12 guage semiauto. 28inch ribbed and vented barrel with modified choke. Gun fires and cycles well. Text for pics.	9mm Beretta hand gun for sale at a very good price with a delivery to any interested buyer in the state.contact me at silven2016@yandex.com or text me at	1956 Mossin nagant. 7.62×54 bolt action Russian war rifle. Missing bayonet but rifle fires and cycles well.	Blue, 9½” barrel, original walnut grips, NcStar 4x32E scope/mount + lighted reticle and lens covers, 2 1/2 lb trigger (professional job), PRO-TECH OUTDOORS cordura shoulder/belt holster, original plastic box/lock & key/.22LR cylinder/factory test envelope w/fired casing/instruction manual/original rear sight/ original trigger spring/protective gun sleeve. Everything appears as new except slight .22WRM cylinder spin mark with only 1 to 1½ 50-round boxes fired. .22LR cylinder never used. Scope is set for 50 yds. using Hornady 45 gr FTX .22WMR ammo. Total package f.o.b. Nashville, TN, \$539. Must have TN driver’s license and/ or TN CCW permit, or delivery to FFL required with photo of Driver’s License and cleared payment.
Expires1366	45 days, 23 hours	This ad has expired	46 days, 4 hours	89 days, 7 hours
Facebook2081	2015	2015	2015	2016
Firearms1276	raquo; Shotguns » Winchester model 140	raquo; Pistols » Semi-Auto » Beretta hand gun for sale	raquo; Rifles » Bolt Action » 1956 Mossin nagant	raquo; Pistols » Revolver » Ruger New Model Single-Six 9 1/2″
Group2122	Non-Felon / Legal	Non-Felon / Legal	Non-Felon / Legal	Mentally Capable / Non-Felon / Legal
ID1421	47955db13ce85f8e	930552dc84196efa	8655da9d2455236	6256c6aa4070496

Extraction Evaluation

10 websites, 5 pages each

fields

	Title	Desc	Seller	Date	Price	Loc	Cat	Member Since	Expires	Views	ID
Perfect	1.0 (50/50)	.76 (37/49)	.95 (40/42)	.83 (40/48)	.87 (39/45)	.51 (23/45)	.68 (34/50)	1.0 (35/35)	.52 (15/29)	.76 (19/25)	.97 (35/36)
Including partial and extra data	1.0 (50/50)	.98 (48/49)	.95 (40/42)	.83 (40/48)	.98 (44/45)	.84 (38/45)	.88 (44/50)	1.0 (35/35)	.55 (16/29)	1.0 (25/25)	1.0 (36/36)

Discussion

- Inferlink approach solves some of the key limitations of Roadrunner
 - Pages do not all have to be of the same type
 - Multiple optionals would be treated as different page types
 - Scales well with complex pages

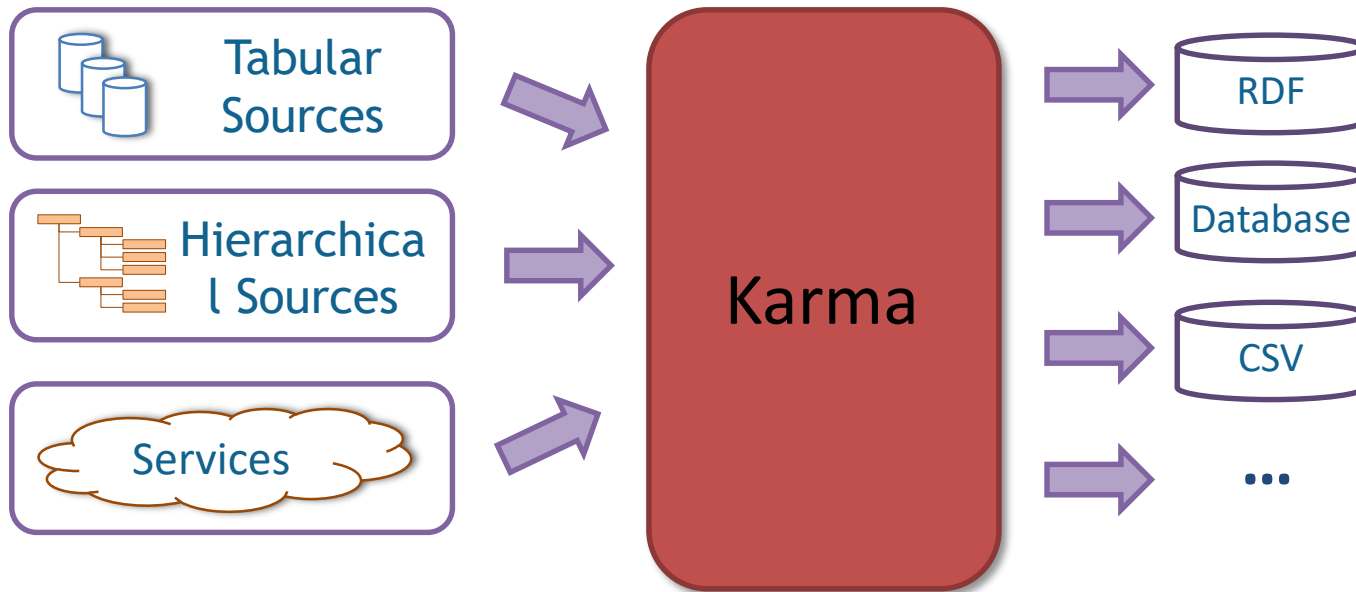
Web Data Extraction Software

- Beautiful Soup
 - <http://www.crummy.com/software/BeautifulSoup/>
 - Python library to manually write wrappers
- Jsoup
 - <http://jsoup.org/>
 - Java library to manually write wrappers
- ScrapingHub
 - <http://scrapinghub.com/>
 - Portia provides a wrapper learner
- Others
 - <https://www.quora.com/Which-are-some-of-the-best-web-data-scraping-tools>
 - Tell us if you find a good one!

Aligning and Integrating Data in Karma

Karma

Interactive tool for rapidly extracting, cleaning, transforming, integrating and publishing data

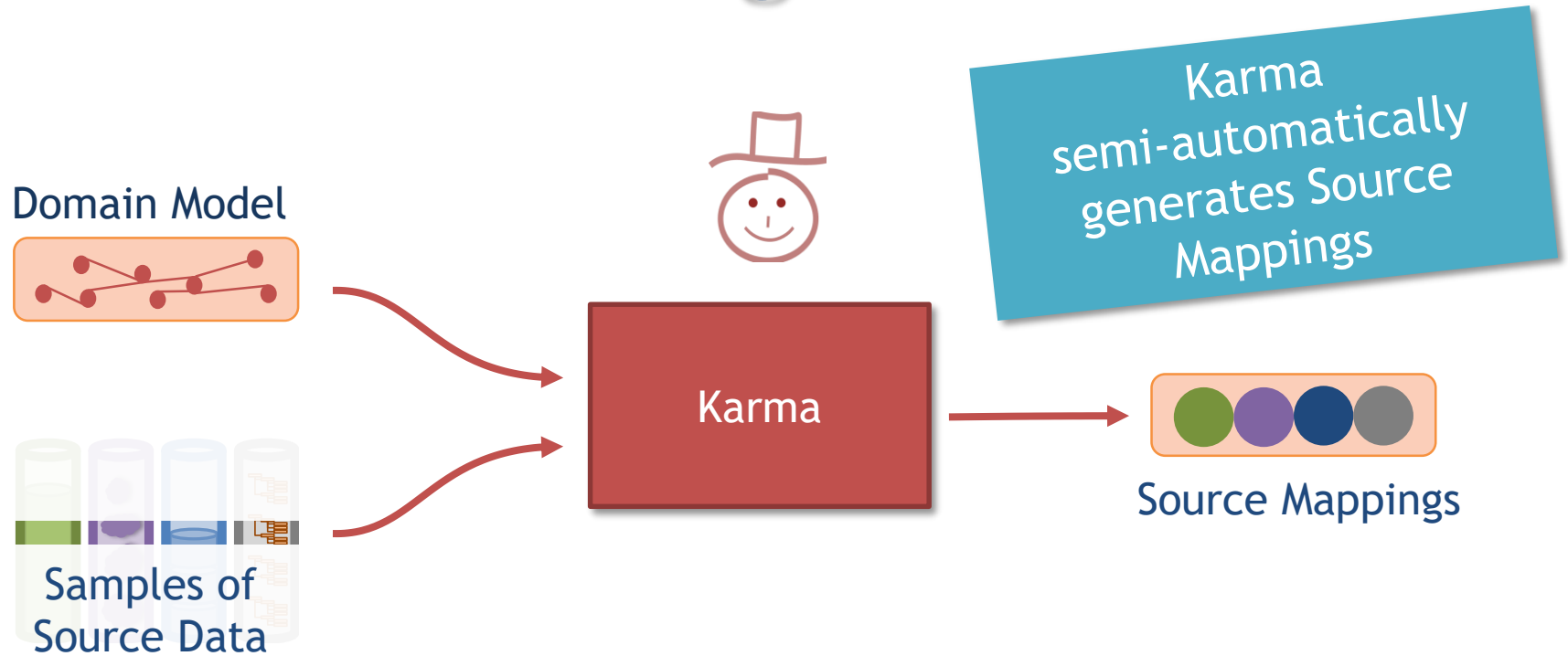


<http://www.isi.edu/integration/karma>

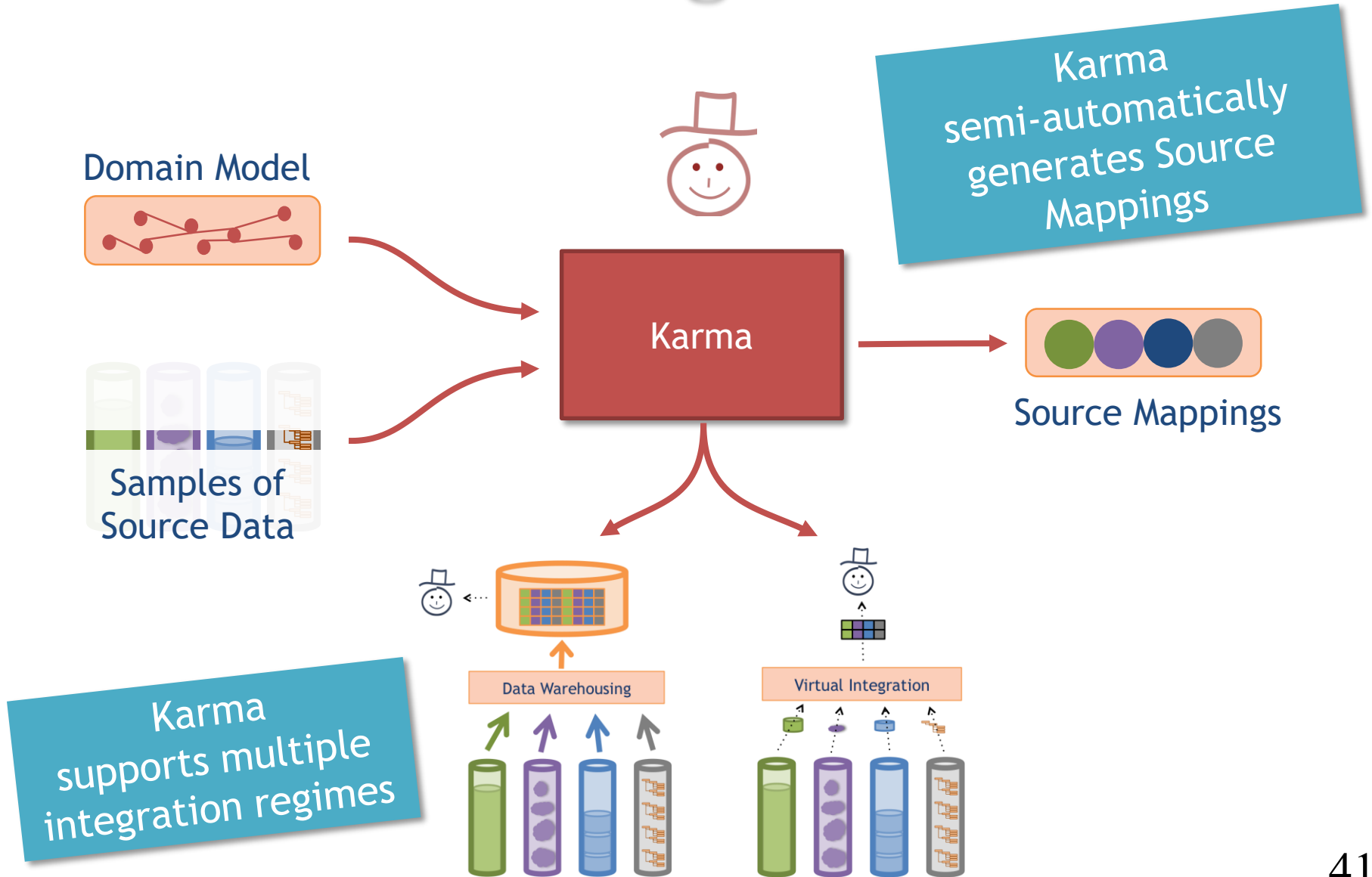


[@KarmaSemWeb](https://twitter.com/KarmaSemWeb)

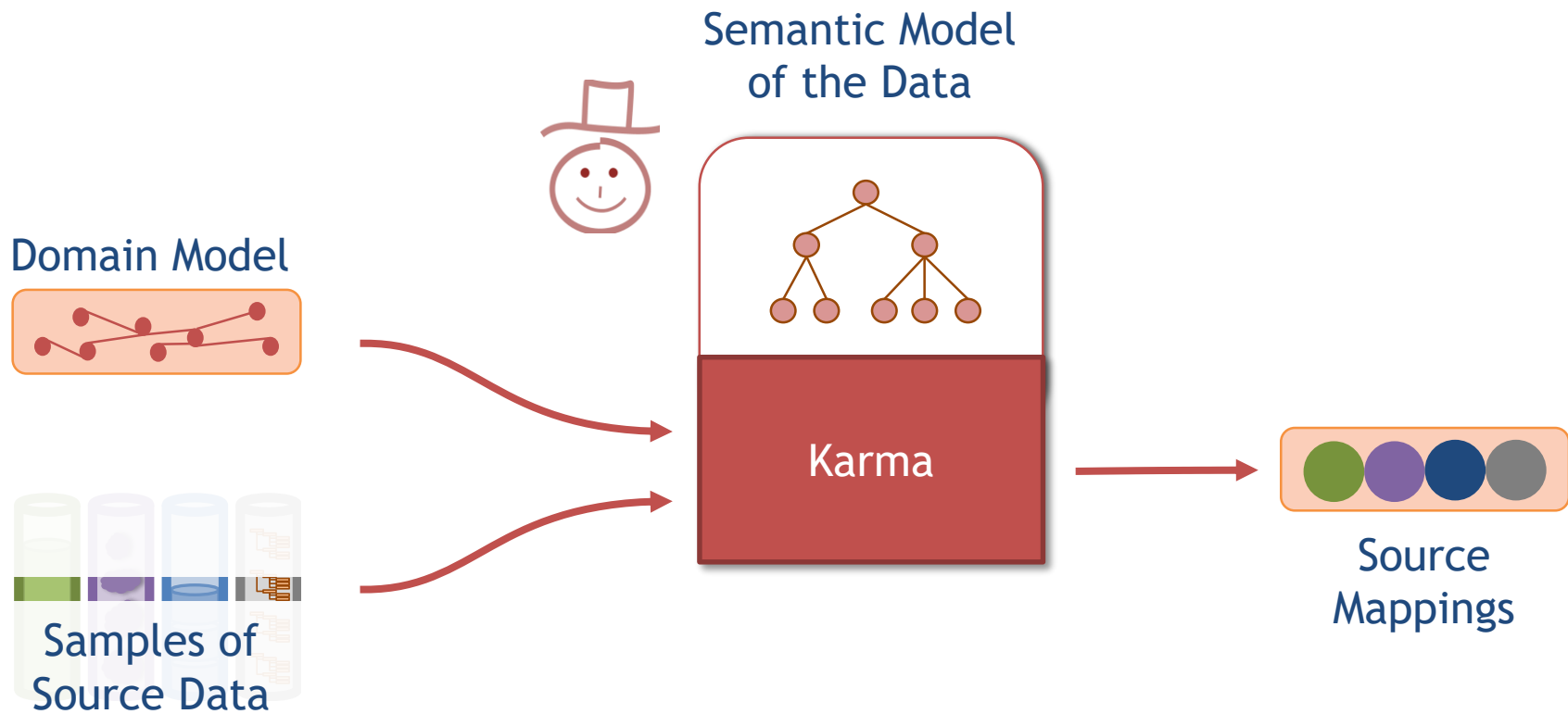
Information Integration in Karma



Information Integration in Karma



Secret Sauce: Karma Understands Your Data



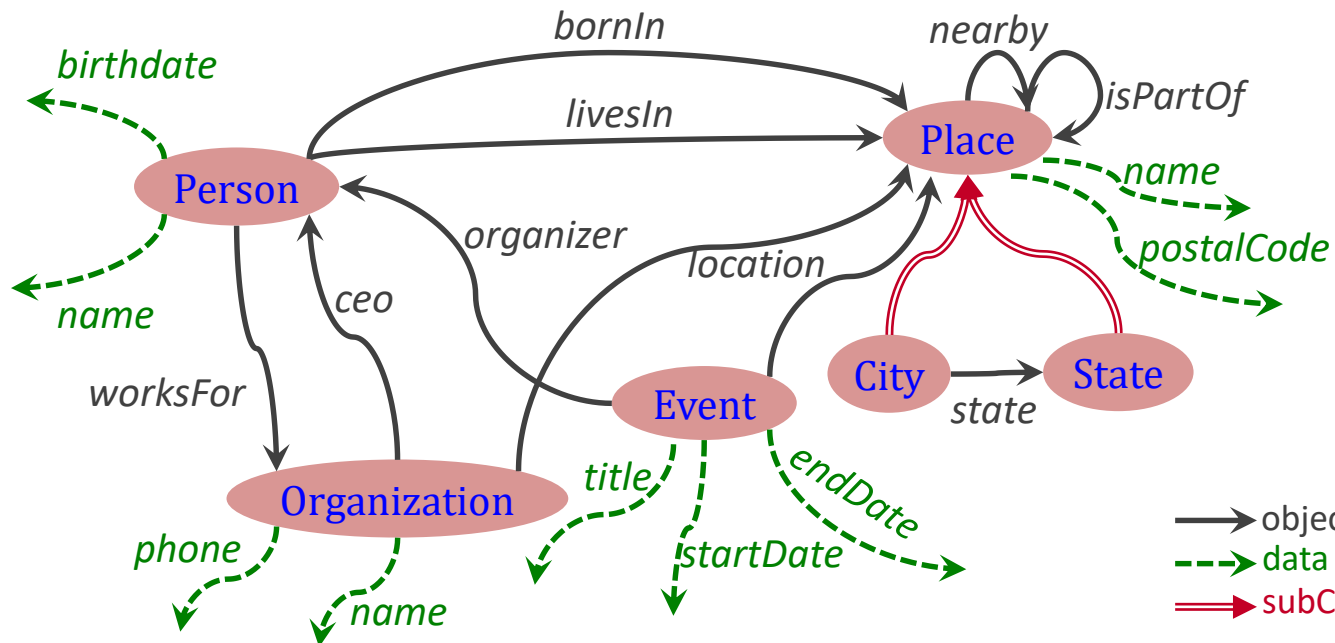
Karma semi-automatically builds a semantic model of your data

What is a Semantic Model?

Describe sources using classes & relationships in an ontology

Source

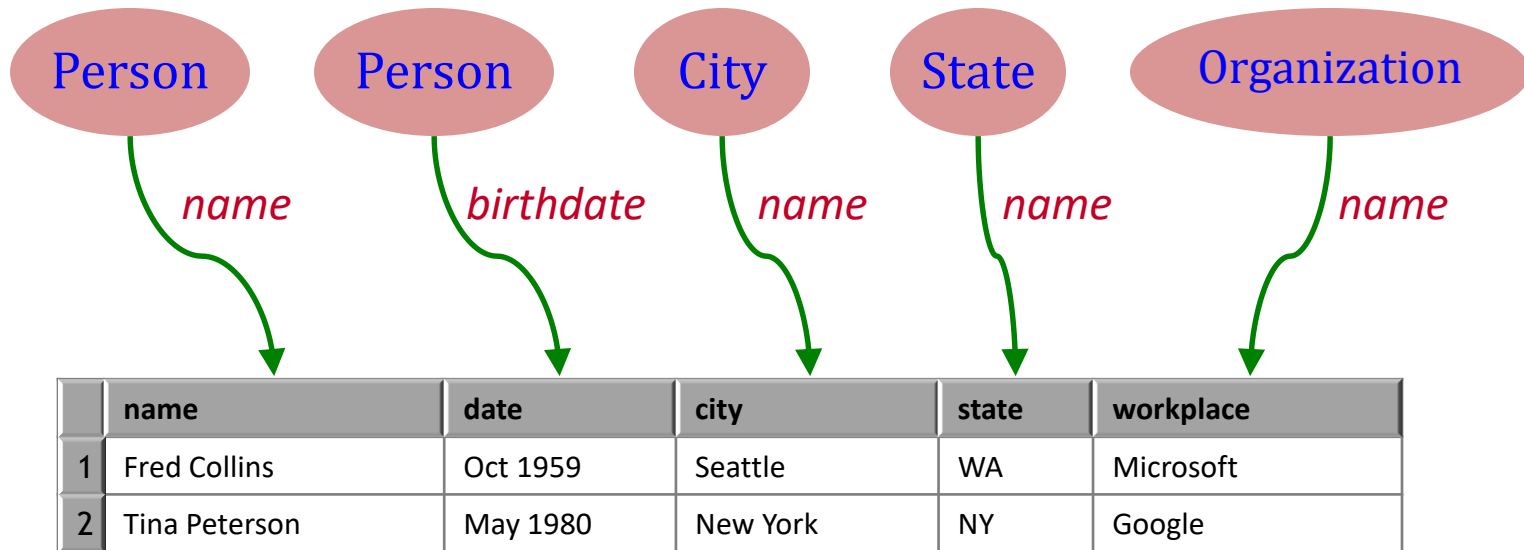
	name	date	city	state	workplace
1	Fred Collins	Oct 1959	Seattle	WA	Microsoft
2	Tina Peterson	May 1980	New York	NY	Google



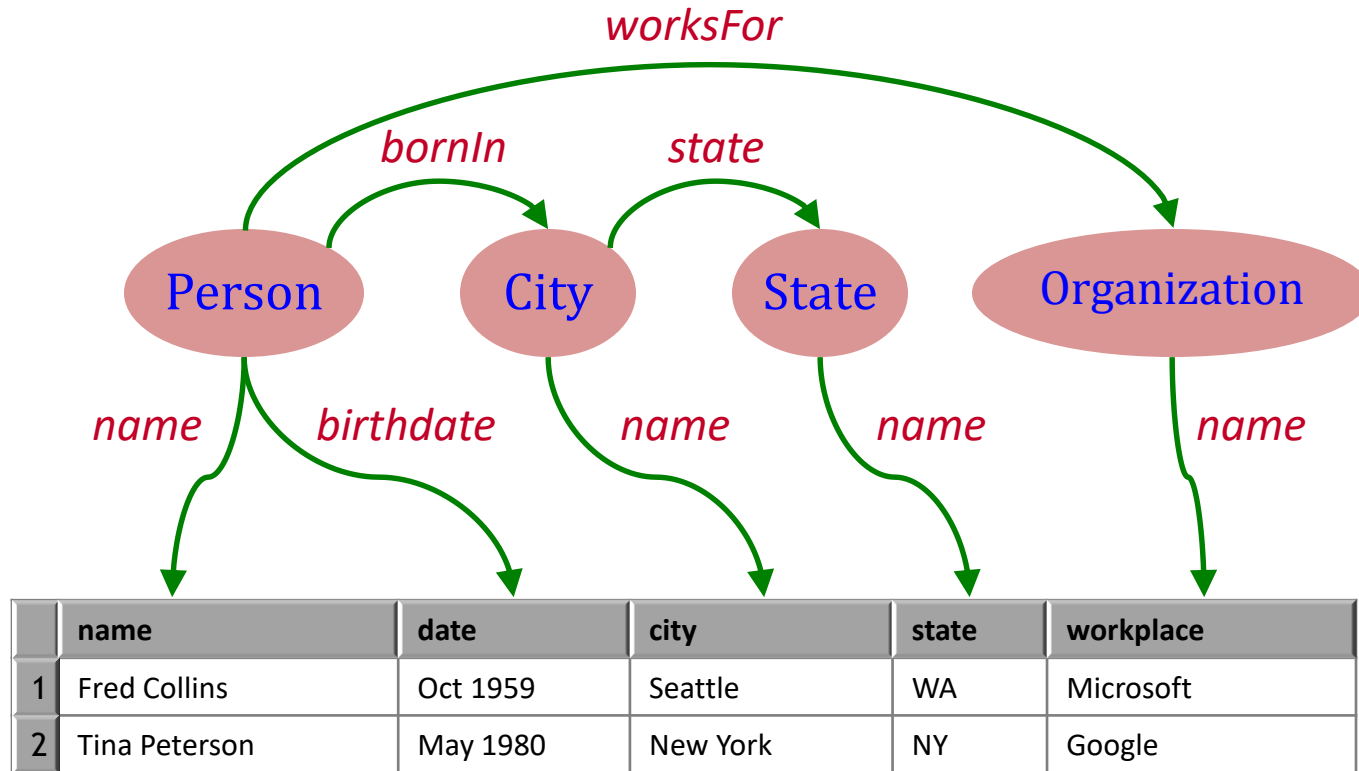
Domain Model

→ object property
 - - - data property
 ==> subClassOf

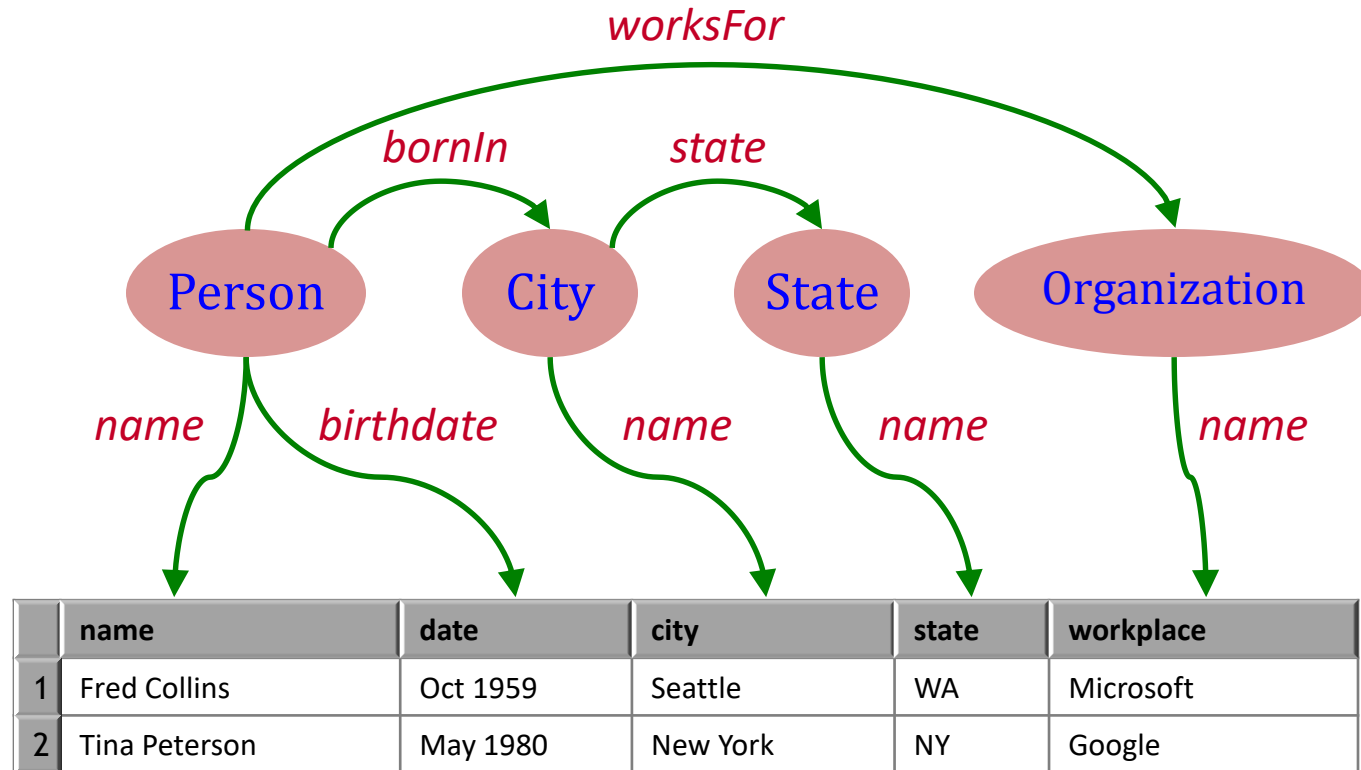
Semantic Types



Relationships

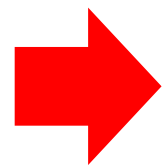
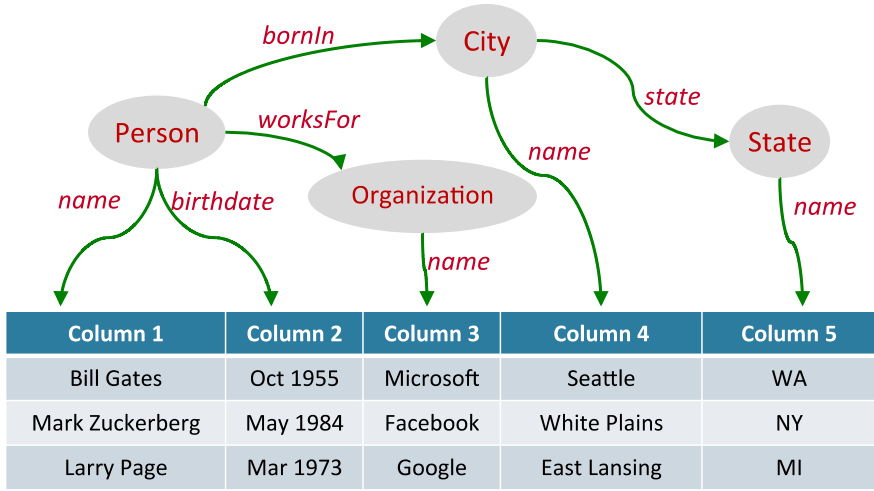


Semantic Model



Semantic models will be formalized as Source Mappings

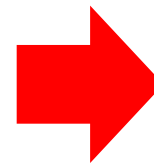
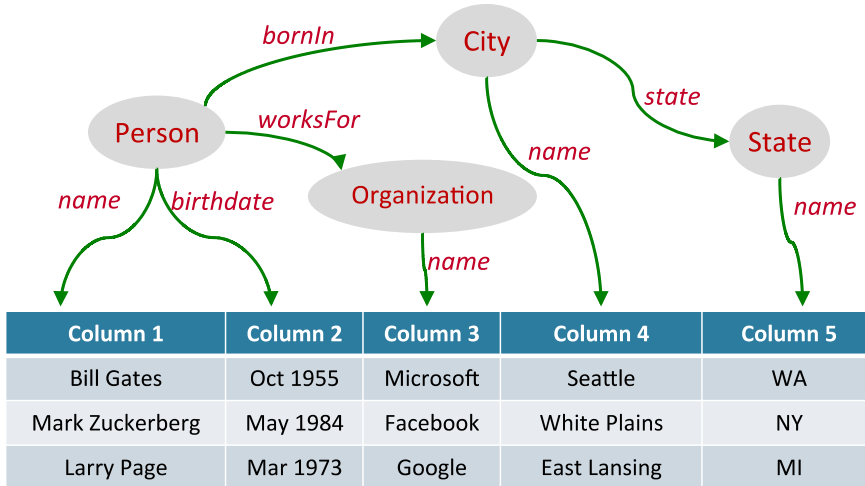
Key ingredient to automate source discovery, data integration, and publishing semantic data (RDF triples)



Knowledge Graphs

Karma uses **semantic models** to **create** knowledge graphs

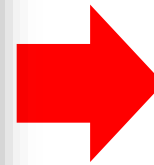
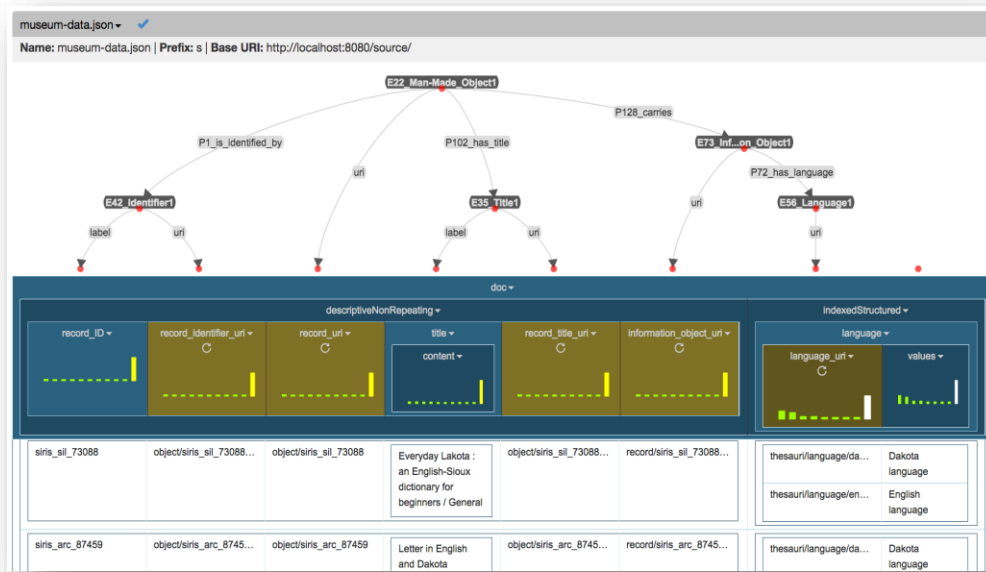
Karma **semi-automatically builds** semantic models



Knowledge
Graphs

Karma uses **semantic models** to **create** knowledge graphs

Karma **semi-automatically builds** semantic models
... and provides a **nice GUI** to edit them



Knowledge
Graphs

Karma uses **semantic models** to **create** knowledge graphs

Semi-automatically Building Semantic Models in Karma

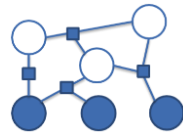
Approach

[Knoblock et al, ESWC 2012]

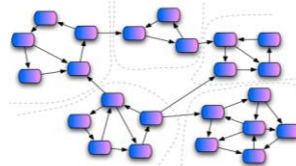


	name	birthdate	city	state	workplace
1	Fred Collins	Oct 1959	Seattle	WA	Microsoft
2	Tina Peterson	May 1980	New York	NY	Google

Sample Data



**Learn
Semantic Types**



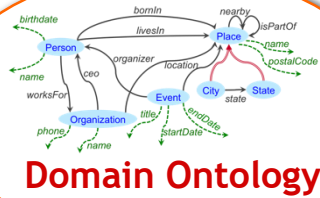
Construct a Graph



**Steiner
Tree**



**Extract
Relationships**



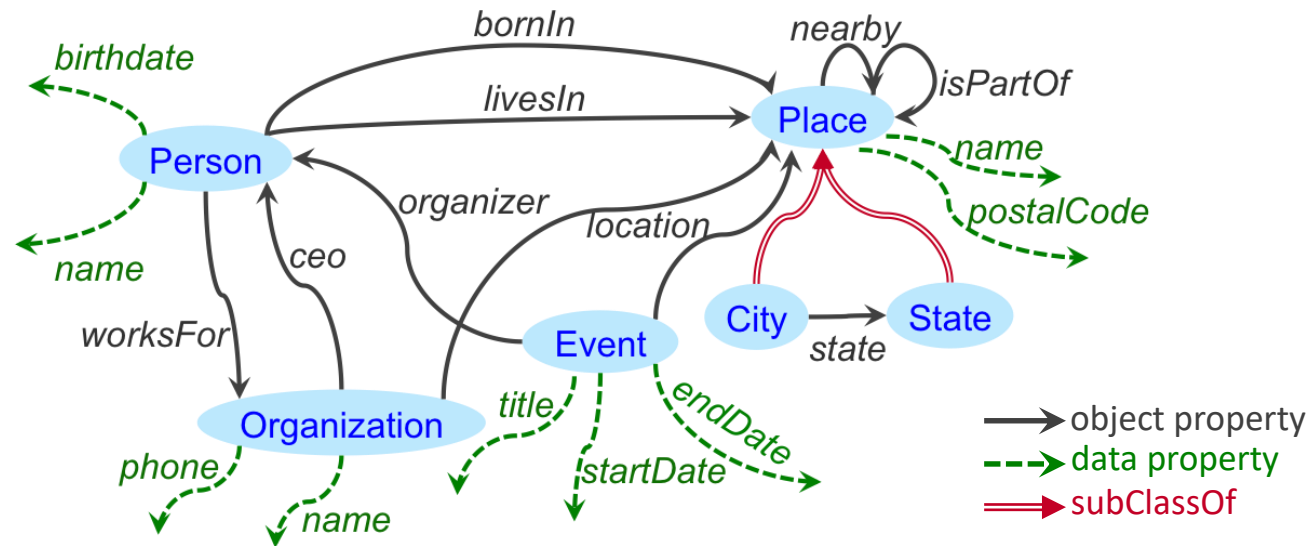
Domain Ontology

Example

Source

	name	date	city	state	workplace
1	Fred Collins	Oct 1959	Seattle	WA	Microsoft
2	Tina Peterson	May 1980	New York	NY	Google

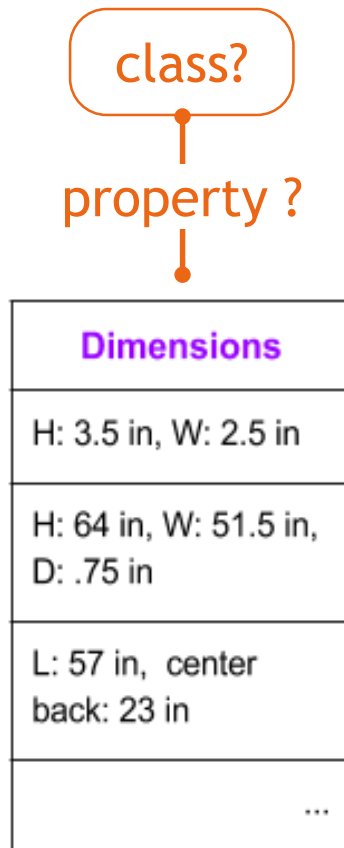
Domain Ontology



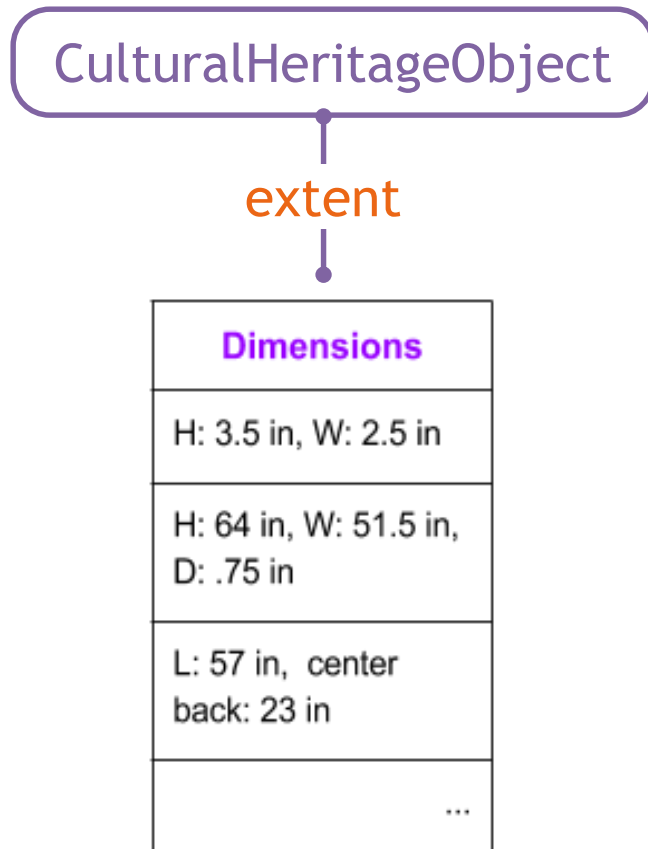
Find a semantic model for the source (map the source to the ontology)

Learning Semantic Types

[Krishnamurthy et al., ESWC 2015]



Learning Semantic Types



1- User specifies

2- System learns

Learning Semantic Types

CulturalHeritageObject

extent

Dimensions
H: 3.5 in, W: 2.5 in
H: 64 in, W: 51.5 in, D: .75 in
L: 57 in, center back: 23 in
...

Extent
52.1 x 71.4 cm (20 1/2 x 28 1/8 in.)
9 3/4 x 7 9/16 in.
H: 19 x W: 15 1/4 x D: 8 1/4 in.
...

Learning Semantic Types

CulturalHeritageObject

extent

Dimensions
H: 3.5 in, W: 2.5 in
H: 64 in, W: 51.5 in, D: .75 in
L: 57 in, center back: 23 in
...

CulturalHeritageObject

extent

Extent
52.1 x 71.4 cm (20 1/2 x 28 1/8 in.)
9 3/4 x 7 9/16 in.
H: 19 x W: 15 1/4 x D: 8 1/4 in.
...

Requirements

- Learn from a small number of examples
- Work on both textual and numeric values
- Learn quickly and highly scalable to large number of semantic types

Approach for Textual Data

- **Document**: each column of data
- **Label**: each semantic type
- Use **Apache Lucene** to index the labeled documents
- Compute **TF/IDF vectors** for documents
- Compare documents using **Cosine Similarity** between TF/IDF vectors

Dimensions
H: 3.5 in, W: 2.5 in
H: 64 in, W: 51.5 in, D: .75 in
L: 57 in, center back: 23 in
...

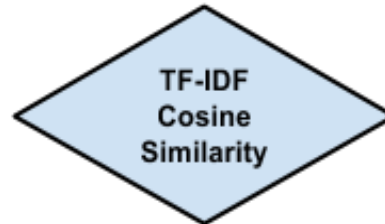
Approach for Textual Data

Dimensions
H: 3.5 in, W: 2.5 in
H: 64 in, W: 51.5 in, D: .75 in
L: 57 in, center back: 23 in
...

Term: TF-IDF score
 h: 0.375
 w: 0.336
 in: 0.491
 centre: 0.241
 back: 0.301

Extent
52.1 x 71.4 cm (20 1/2 x 28 1/8 in.)
9 3/4 x 7 9/16 in.
H: 19 x W: 15 1/4 x D: 8 1/4 in.
...

Term: TF-IDF score
 h: 0.414
 w: 0.364
 d: 0.245
 cm: 0.354
 in: 0.395



$$tf(t, d) = frequency^{1/2}$$

$$idf(t) = 1 + \log\left(\frac{numDocs}{docFreq + 1}\right)$$

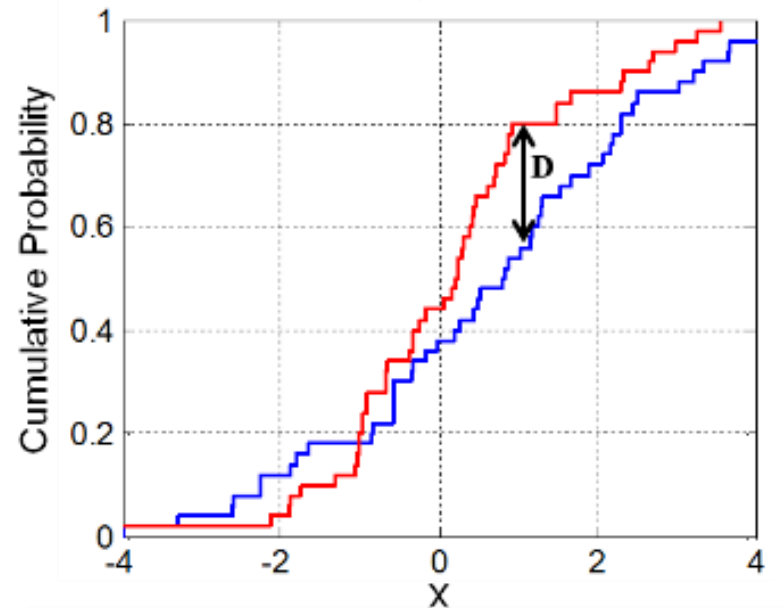
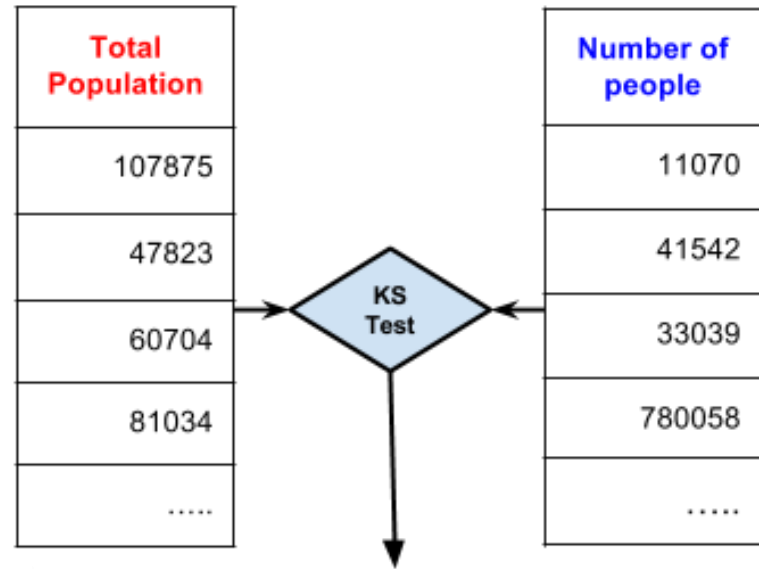
$$sim(q, d) = \frac{V(q) \cdot V(d)}{|V(q)| \cdot |V(d)|}$$

Approach for Numeric Data

- **Distribution** of values in different semantic types is different, e.g., temperature vs. population
- Use **Statistical Hypothesis Testing** to see which distribution fits best
- Welch's T-test, Mann-Whitney U-test and **Kolmogorov-Smirnov Test**

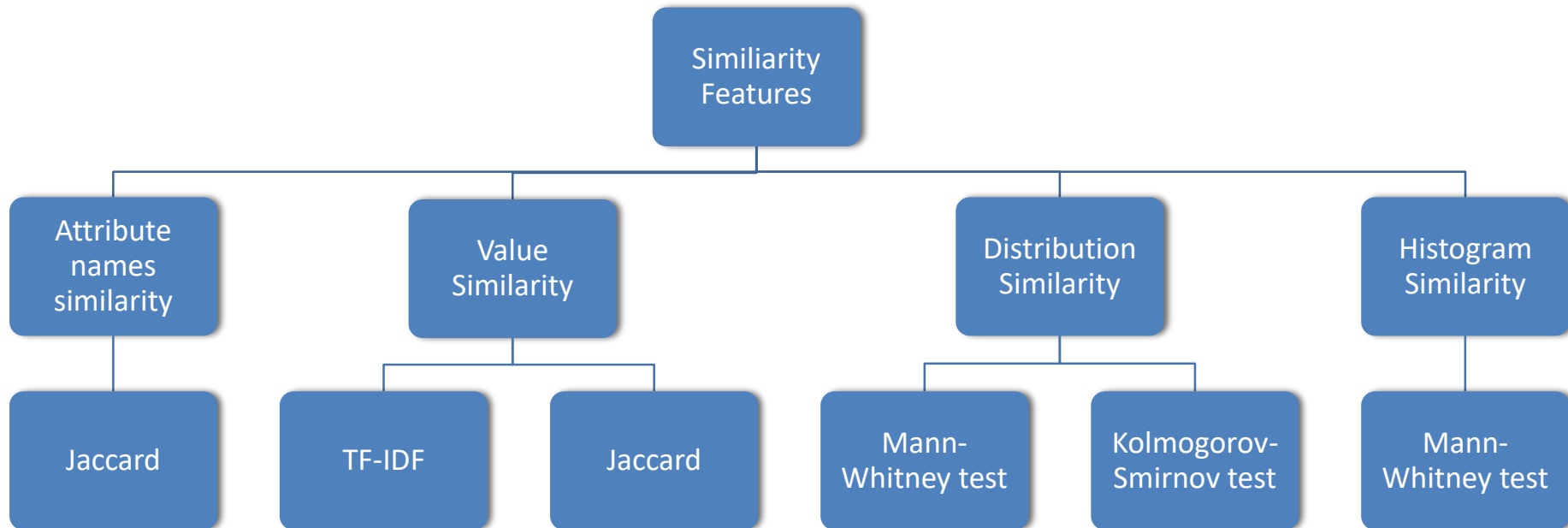
Total Population	Number of people
107875	11070
47823	41542
60704	33039
81034	780058
.....

Approach for Numeric Data



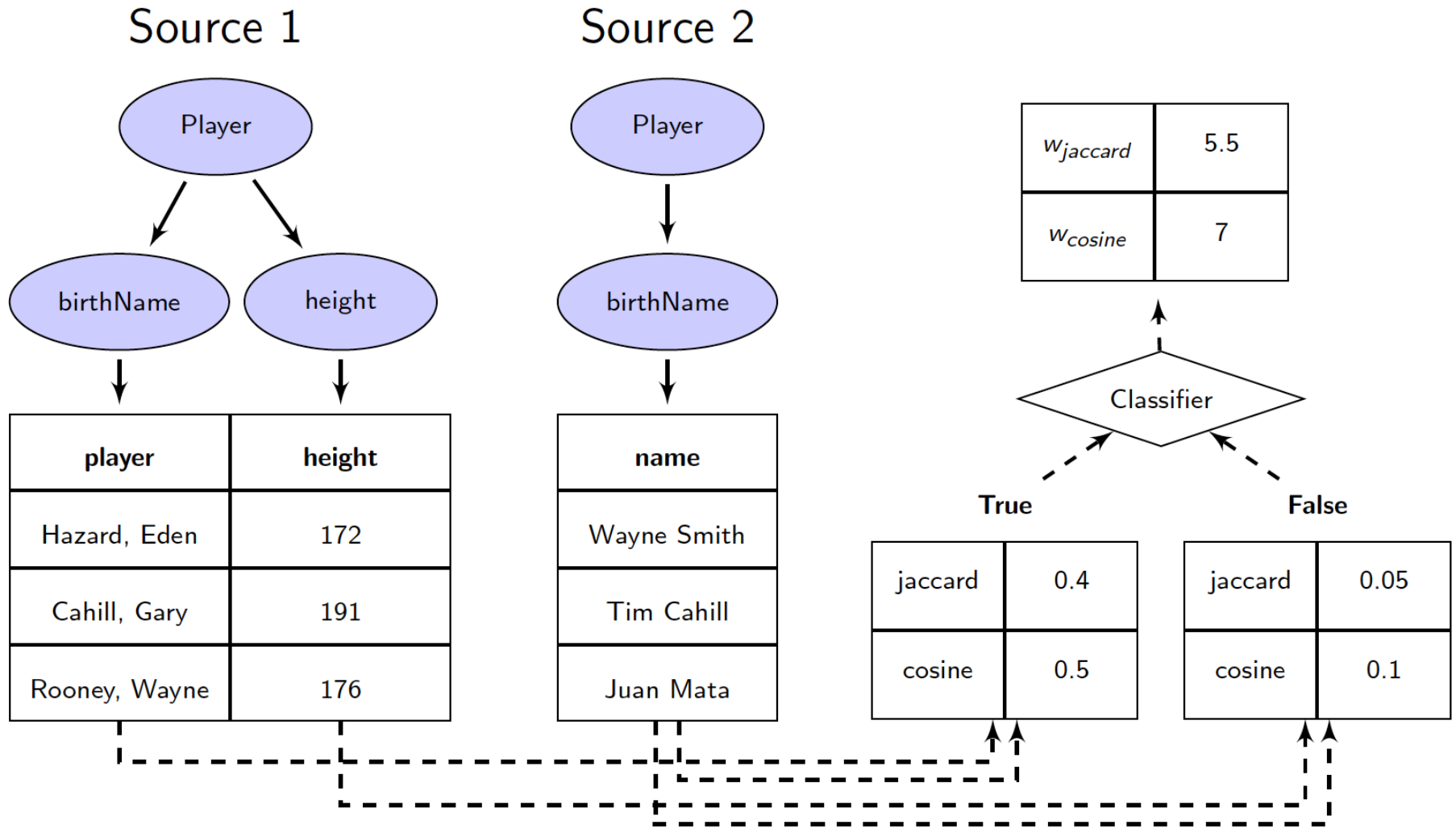
$$D_{N_1, N_2} = \sup_x |F_{1, N_1}(x) - F_{2, N_2}(x)|$$

Similarity features

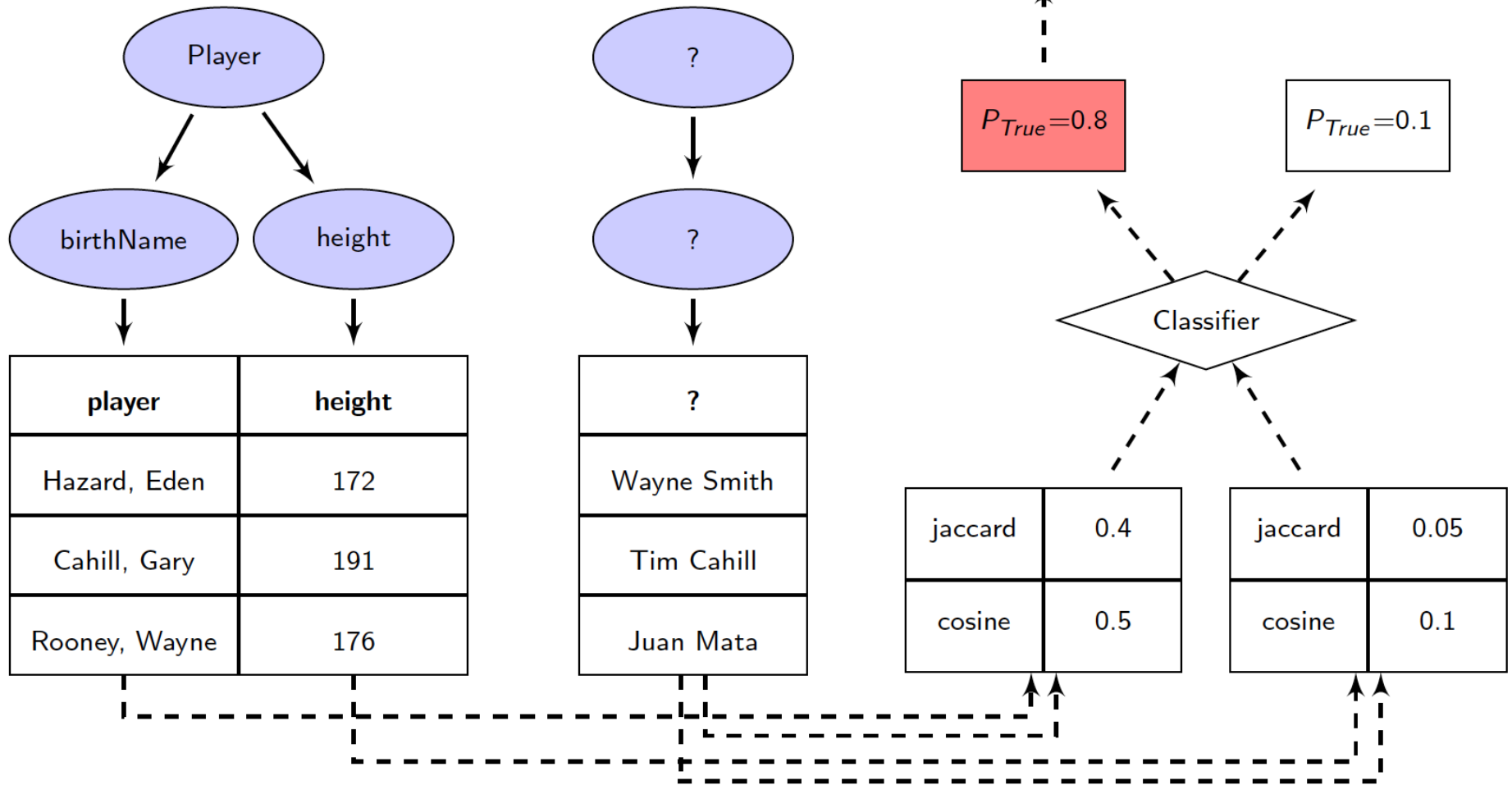


Training machine learning model

[Pham et al., ISWC 2016]



Predicting new attribute



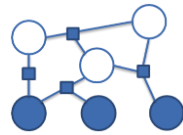
Approach

[Knoblock et al, ESWC 2012]

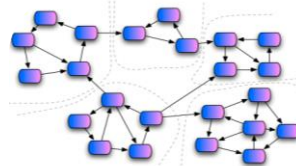


	name	birthdate	city	state	workplace
1	Fred Collins	Oct 1959	Seattle	WA	Microsoft
2	Tina Peterson	May 1980	New York	NY	Google

Sample Data



Learn
Semantic Types



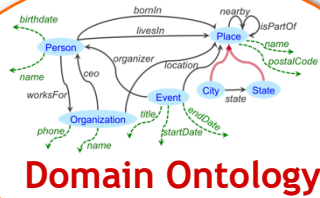
Construct a Graph



Steiner
Tree



Extract
Relationships

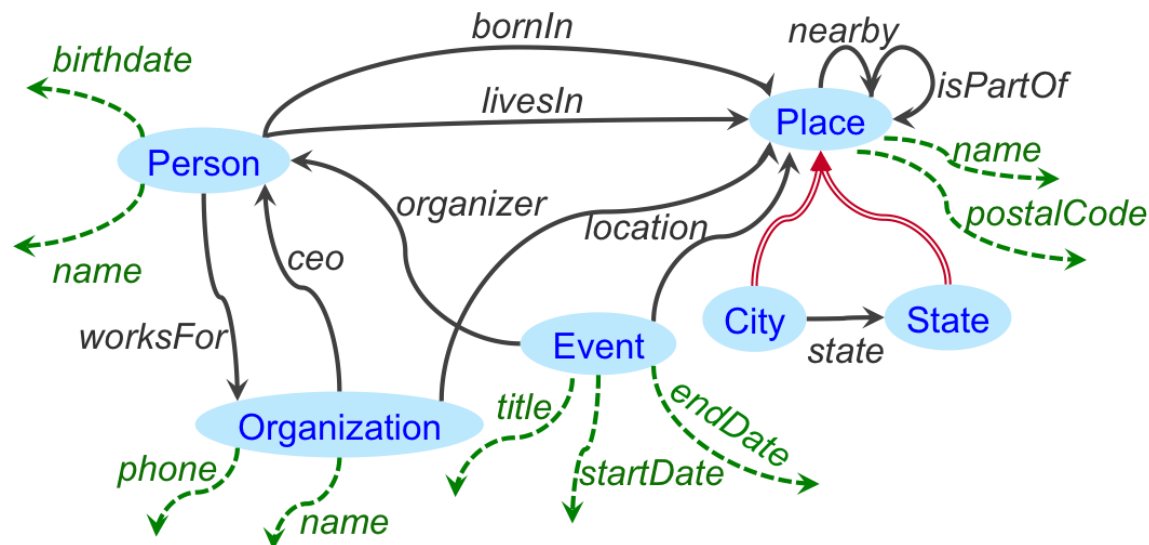


Domain Ontology

Construct a Graph

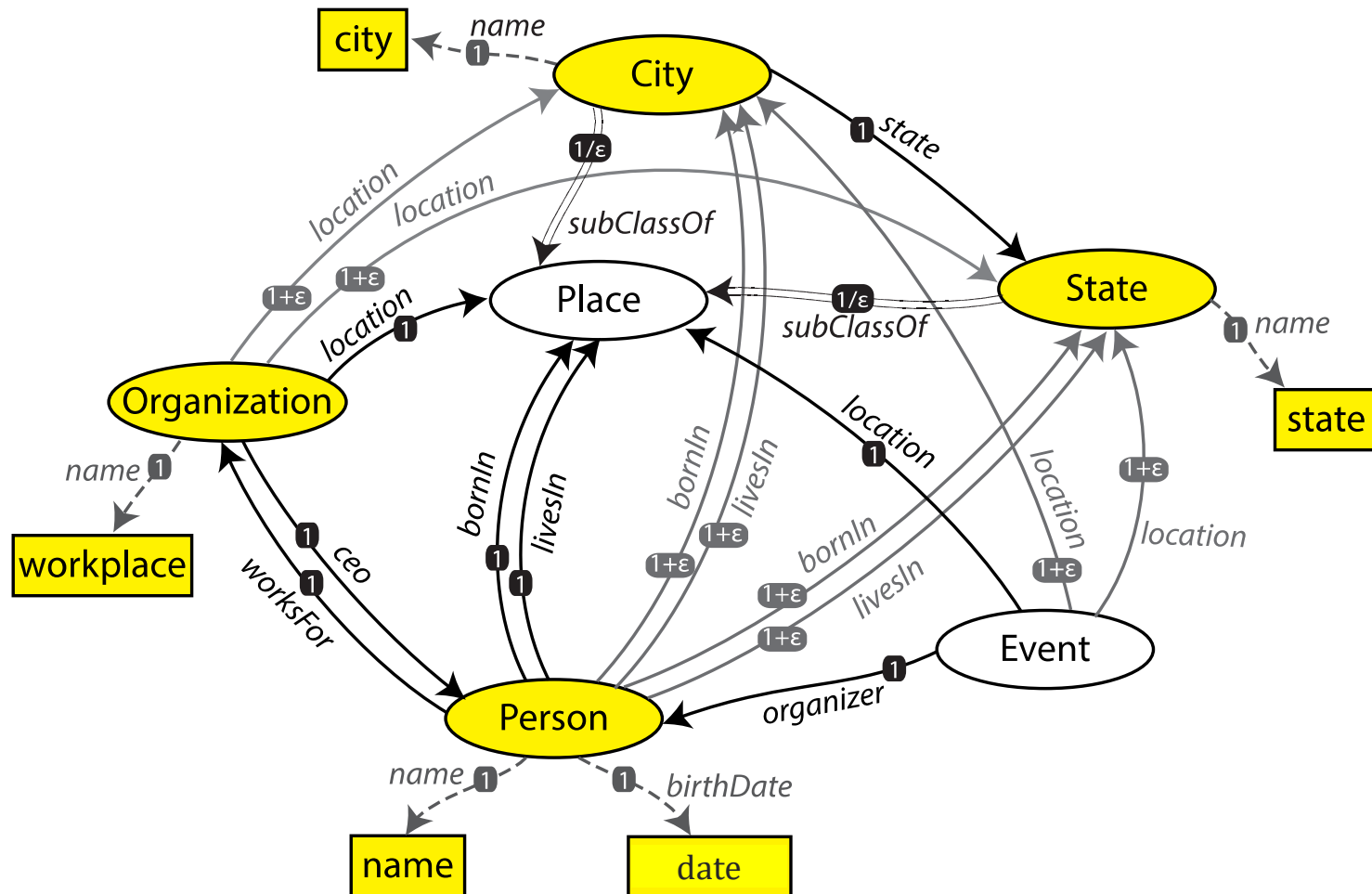
Construct a graph from semantic types and ontology

	name	date	city	state	workplace
1	Fred Collins	Oct 1959	Seattle	WA	Microsoft
2	Tina Peterson	May 1980	New York	NY	Google



Construct a Graph

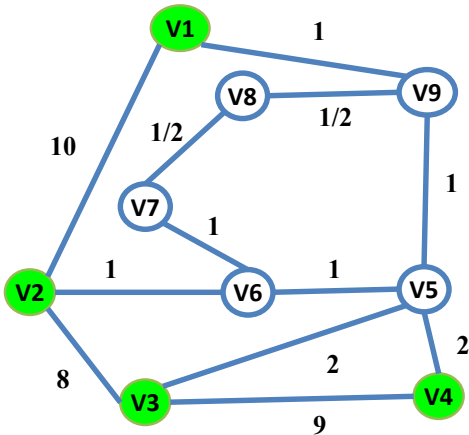
Construct a graph from semantic types and ontology



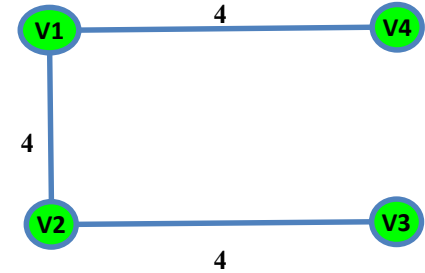
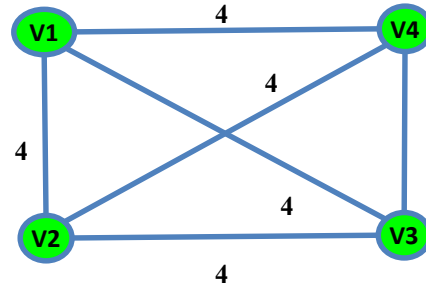
Inferring the Relationships

- Search for minimal explanation
- Steiner tree connecting semantic types over ontology graph
 - Given graph $G=(V,E)$, nodes $S \subset V$, cost $c: E \rightarrow \mathbb{R}$
 - Find a tree of G that spans S with minimal total cost
 - Unfortunately, NP-complete
- Approximation Algorithm [Kou et al., 1981]
 - Worst-case time complexity: $O(|V|^2|S|)$
 - Approximation Ratio: less than 2

Steiner Tree

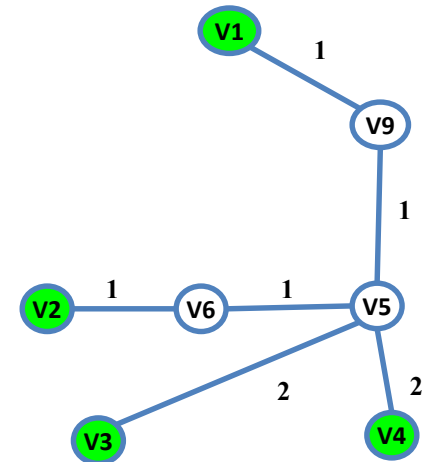
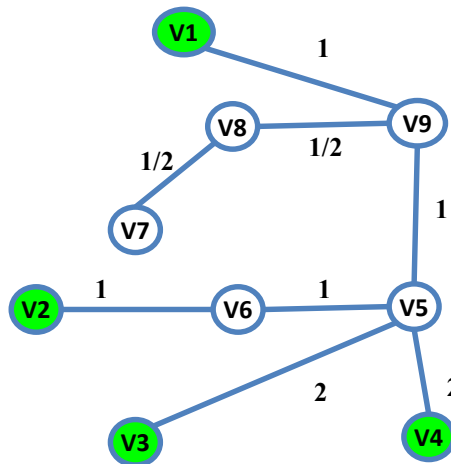
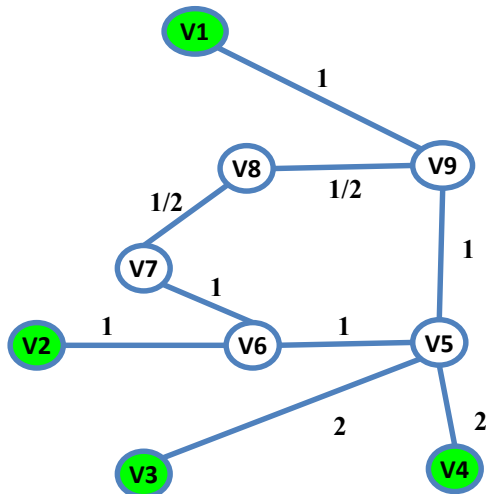


Steiner nodes: {V1, V2, V3, V4}



1. construct the complete graph (Nodes: Steiner Nodes, Links Weights: shortest path from each pair in original G)

2. Compute MST



3. replace each link with the corresponding shortest path in original G

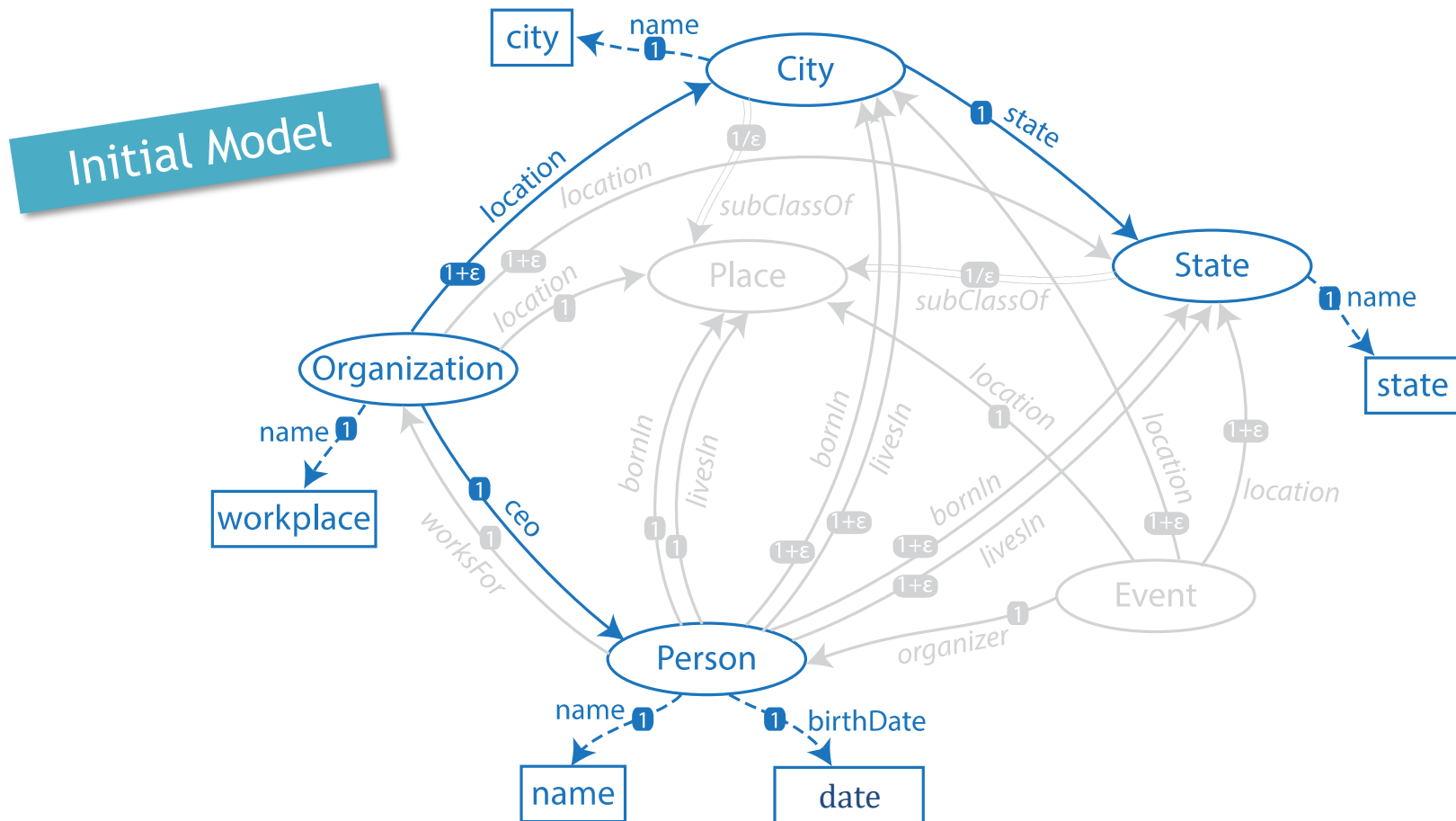
4. Compute MST

5. remove extra links until all leaves are Steiner nodes

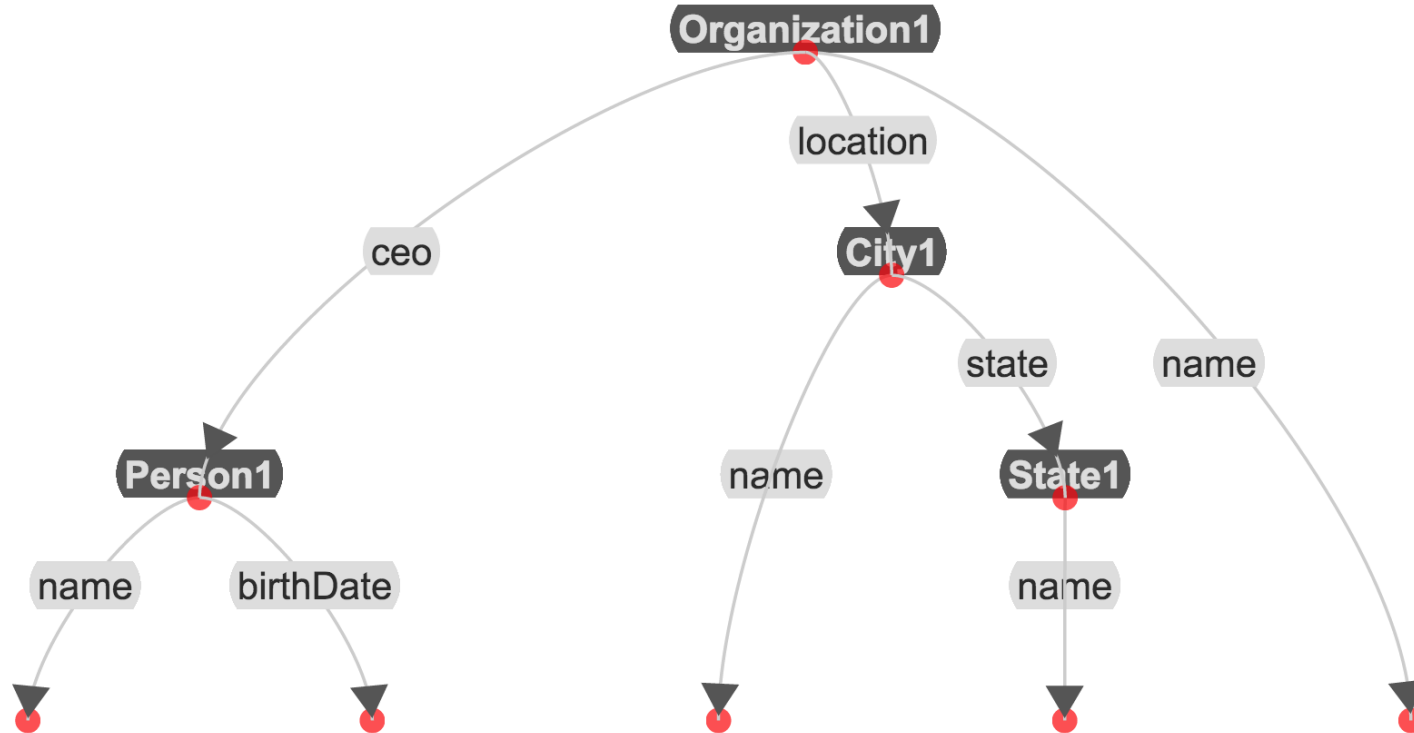
Inferring the Relationships

Select minimal tree that connects all semantic types

- A customized **Steiner tree algorithm**



Result in Karma



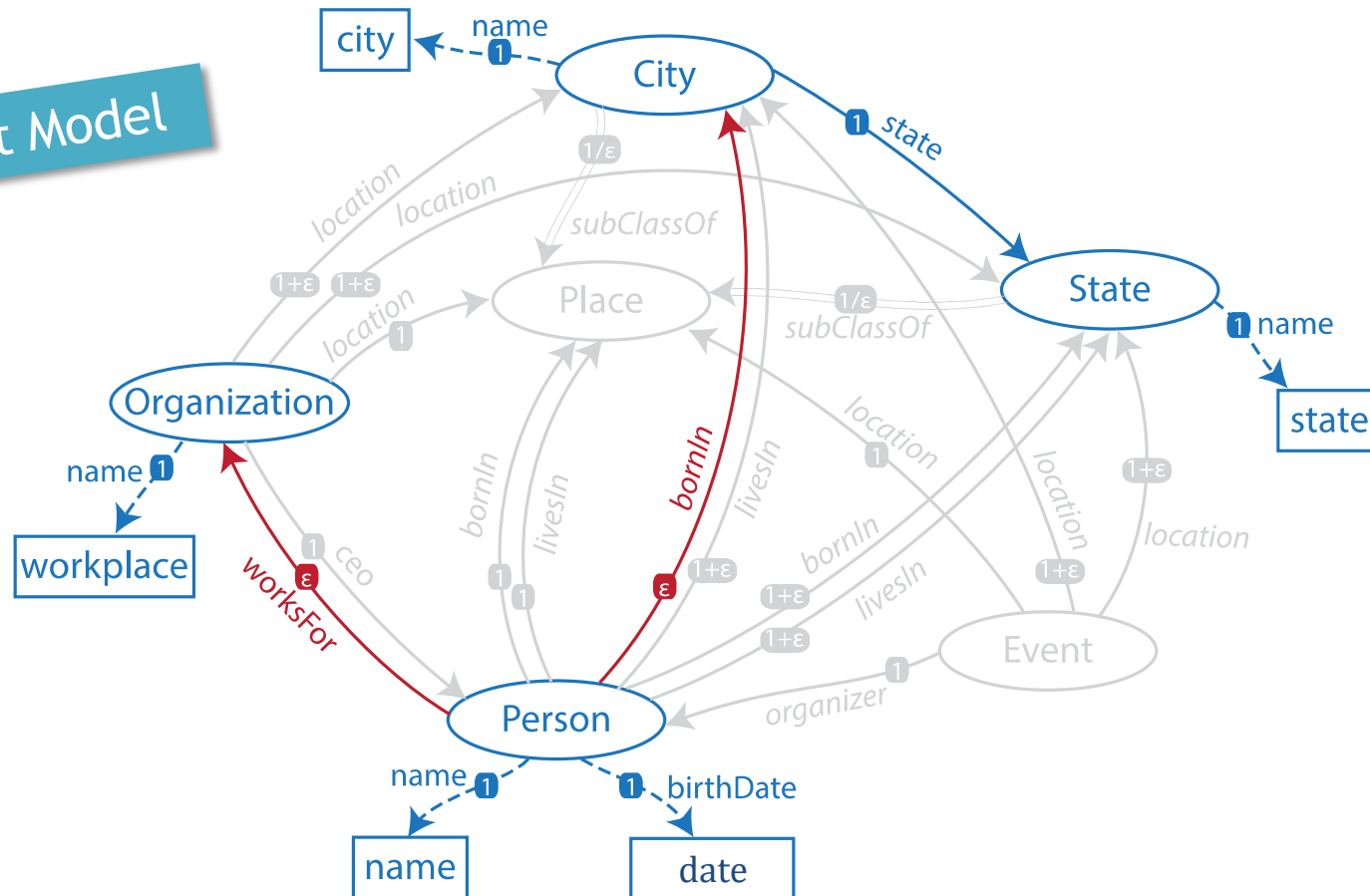
name ▾	birthdate ▾	city ▾	state ▾	workplace ▾
Fred Collins	Oct 1959	Seattle	WA	Microsoft
Tina Peterson	May 1980	New York	NY	Google
Richard Smith	Feb 1975	Los Angeles	CA	Apple

Refining the Model

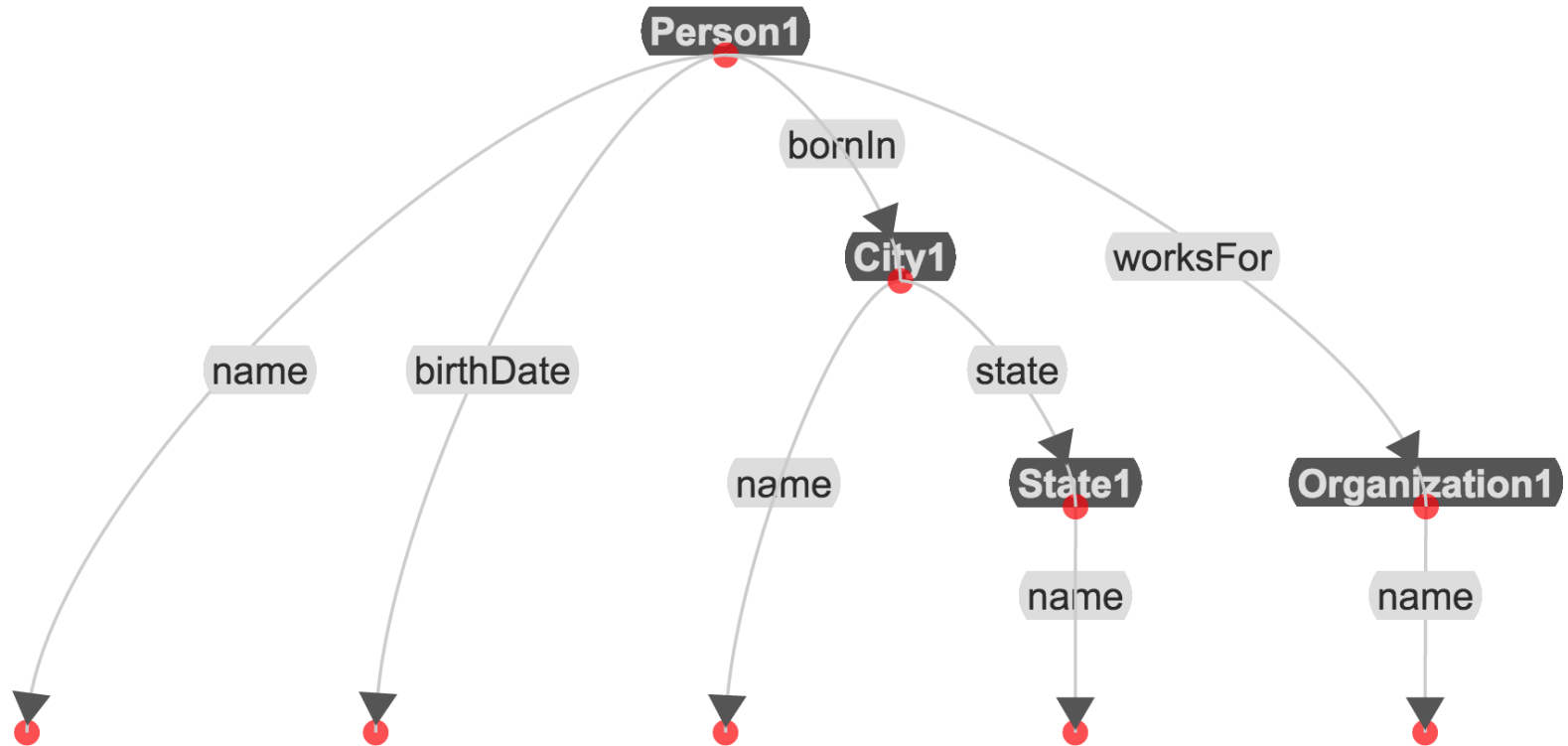
Impose constraints on Steiner Tree Algorithm

- Change weight of selected links to ϵ
- Add source and target of selected link to Steiner nodes

Correct Model



Final Semantic Model



name ▾	birthdate ▾	city ▾	state ▾	workplace ▾
Fred Collins	Oct 1959	Seattle	WA	Microsoft
Tina Peterson	May 1980	New York	NY	Google
Richard Smith	Feb 1975	Los Angeles	CA	Apple

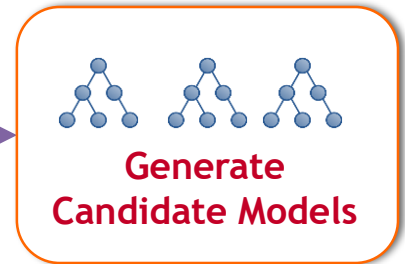
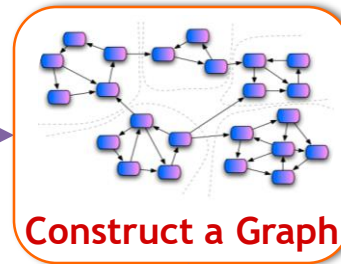
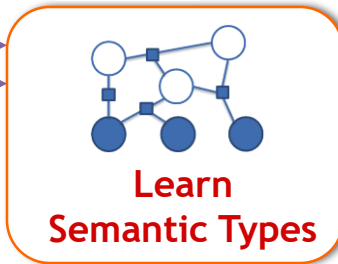
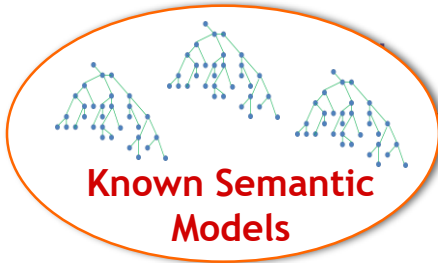
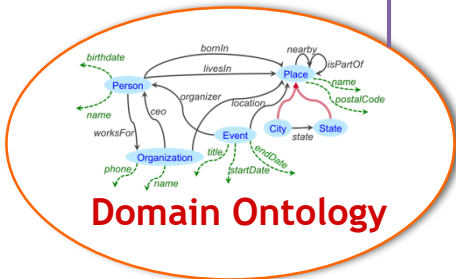
Karma Learns the Source Models

Taheriyani et al., ISWC 2013, ICSC 2014



	name	birthdate	city	state	workplace
1	Fred Collins	Oct 1959	Seattle	WA	Microsoft
2	Tina Peterson	May 1980	New York	NY	Google

Sample Data

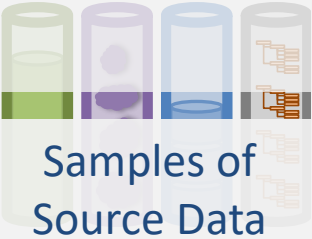


Karma Use Cases



Source Mapping Phase

Domain Model



Domain Expert



Source Mappings

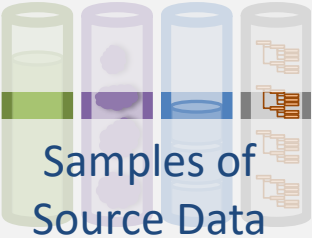


Mapping Phase



Source Mapping and Query Time

Domain Model



Domain Expert

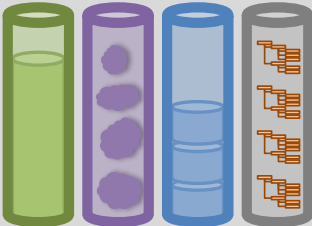


Source Mappings



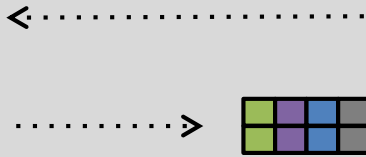
Mapping Phase

Query Phase



Data Warehousing
Virtual Integration

Query



Analyst



VIVO

- [VIVO](#) is a system to build researcher networks across institutions
- Used Karma to map the data about USC faculty to VIVO ontology and publish it as RDF
- VIVO ingest the RDF data
- [Video](#)



An interdisciplinary network

Enabling [collaboration](#) and discovery among [scientists](#) across all disciplines.

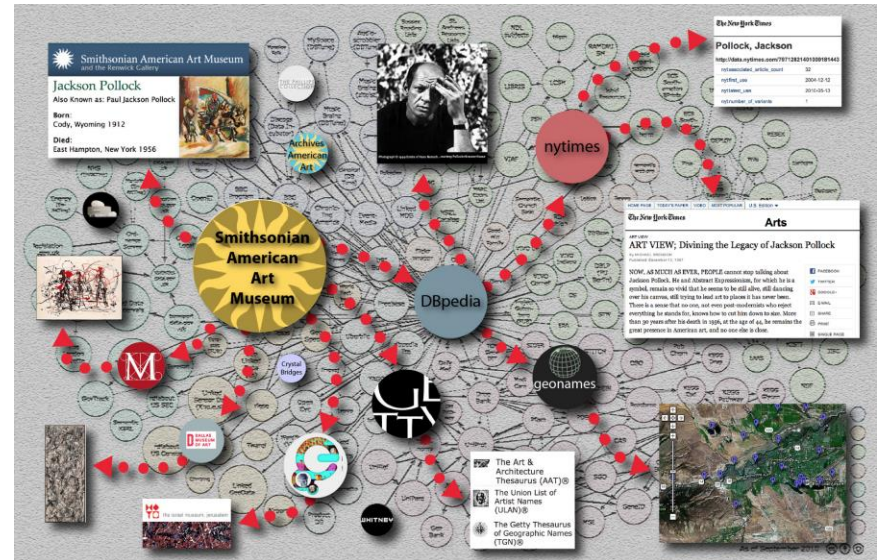
The network of scientists will facilitate scholarly discovery. Institutions will participate in the network by installing VIVO, or by providing semantic web-compliant data to the network.



American Art Collaborative

[Knoblock et al., ISWC 2017]

- Used Karma to convert data of 13 American Art Museums to Linked Open Data
- Modeled according to CIDOC-CRM Ontology
- Linked the generated RDF to DBPedia and ULAN
- [Video](#)



Using Karma to map museum data to the CIDOC CRM ontology

https://www.youtube.com/watch?v=h3_yiBhAJlc

Discussion

- Automatically build rich semantic descriptions of data sources
- Exploit the background knowledge from (i) the domain ontology, and (ii) the known source models
- Semantic descriptions are the key ingredients to automate many tasks, e.g.,
 - Source Discovery
 - Data Integration
 - Service Composition

More Info

karma.isi.edu