USC Viterbi
School of Engineering

Building Semantic Descriptions of Sources

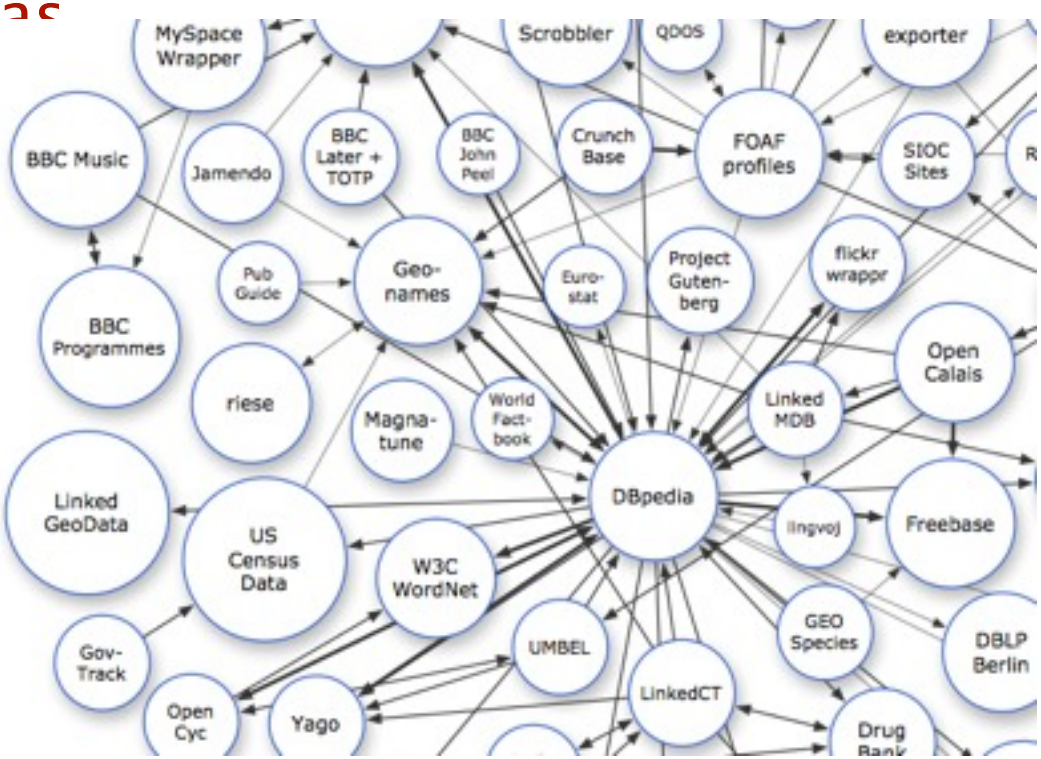
Craig Knoblock
University of Southern California

USC

USC Viterbi School of Engineering

Linked Open Data and Services

- Vast collection of interlinked information
- Various sources and services with different schemas



USC

USC Viterbi School of Engineering

Where do the Semantics Come From?

- Linked Open Data
  - Populated by manually linking or writing procedures that define the links across sources
  - But we don't know how the sources are related
  - In many cases there is no or very limited semantic descriptions of sources
- Linked Open Services
  - Manually constructed or built by wrapping existing Web services
  - Constructing the lifting and lowering rules that relate the services to existing ontologies is a difficult task
  - Even when done, it may only provide a partial description
    - e.g., descriptions of the inputs and outputs, but not the function of a service

USC

USC Viterbi School of Engineering

Outline of the Talk

- Linked Open Data
  - Building and linking ontologies of linked data
- Linked Open Services
  - Building semantic web services from the Deep Web
- Discussion
  - Remaining challenges

USC

USC Viterbi School of Engineering

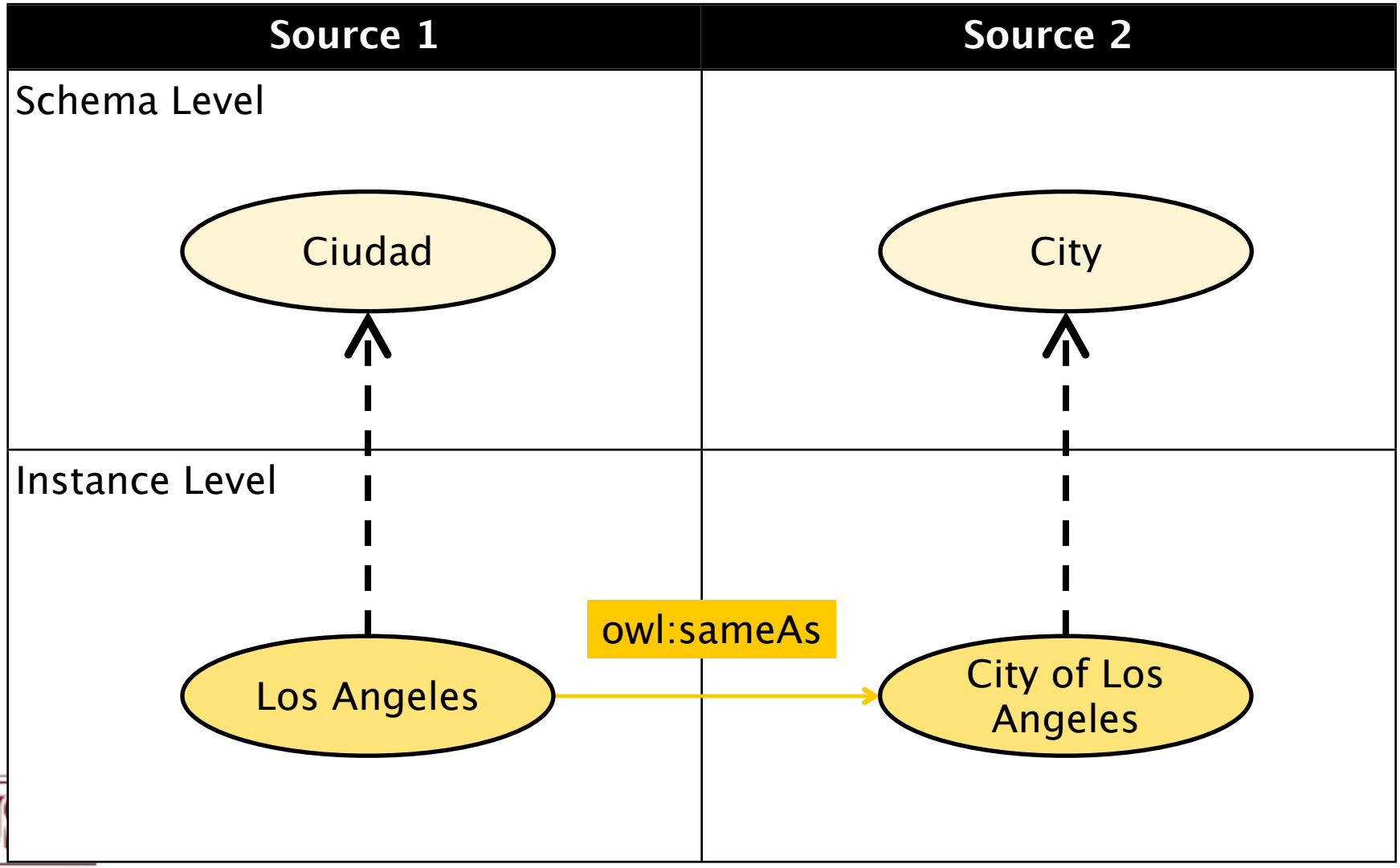
Outline of the Talk

- Linked Open Data
  - Building and linking ontologies of linked data
- Linked Open Services
  - Building semantic web services from the Deep Web
- Discussion
  - Remaining challenges

USC

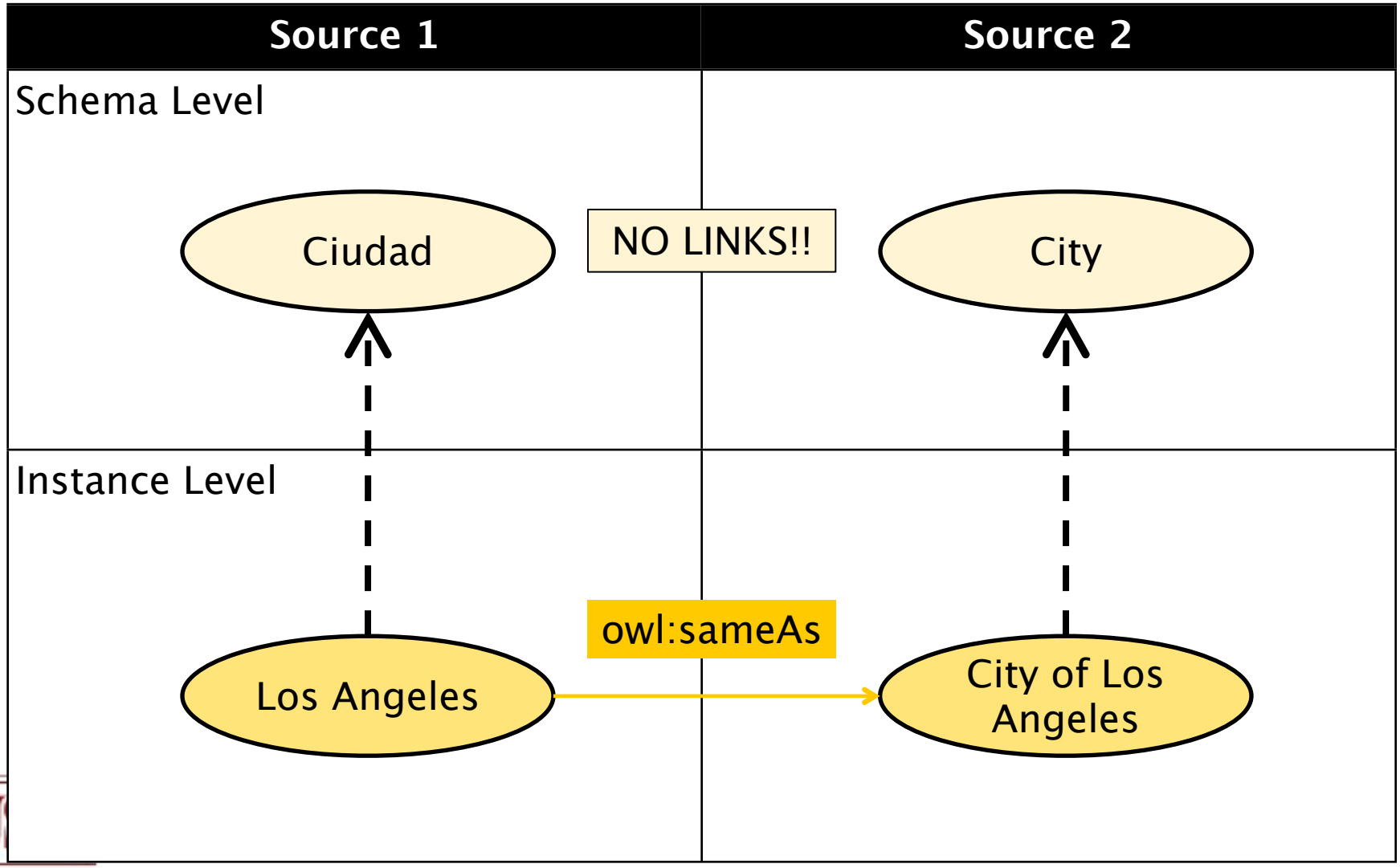
USC Viterbi School of Engineering

Building and linking ontologies of linked data [Parundekar et al., ISWC 2010]



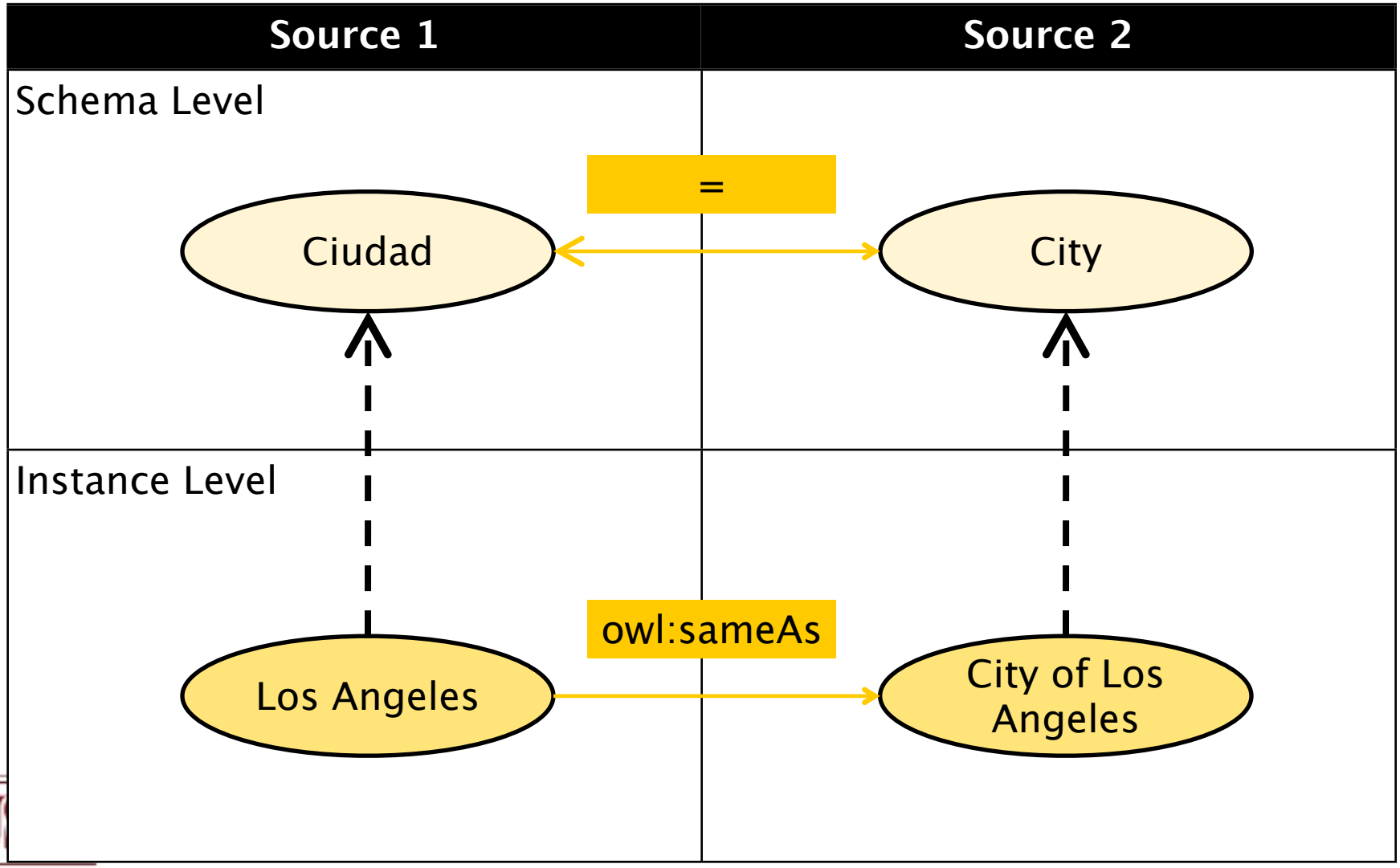
USC Viterbi School of Engineering

Disjoint Schemas



USC Viterbi School of Engineering

Objective 1: Find Schema Alignments



USC Viterbi School of Engineering

Ontologies of Linked Data

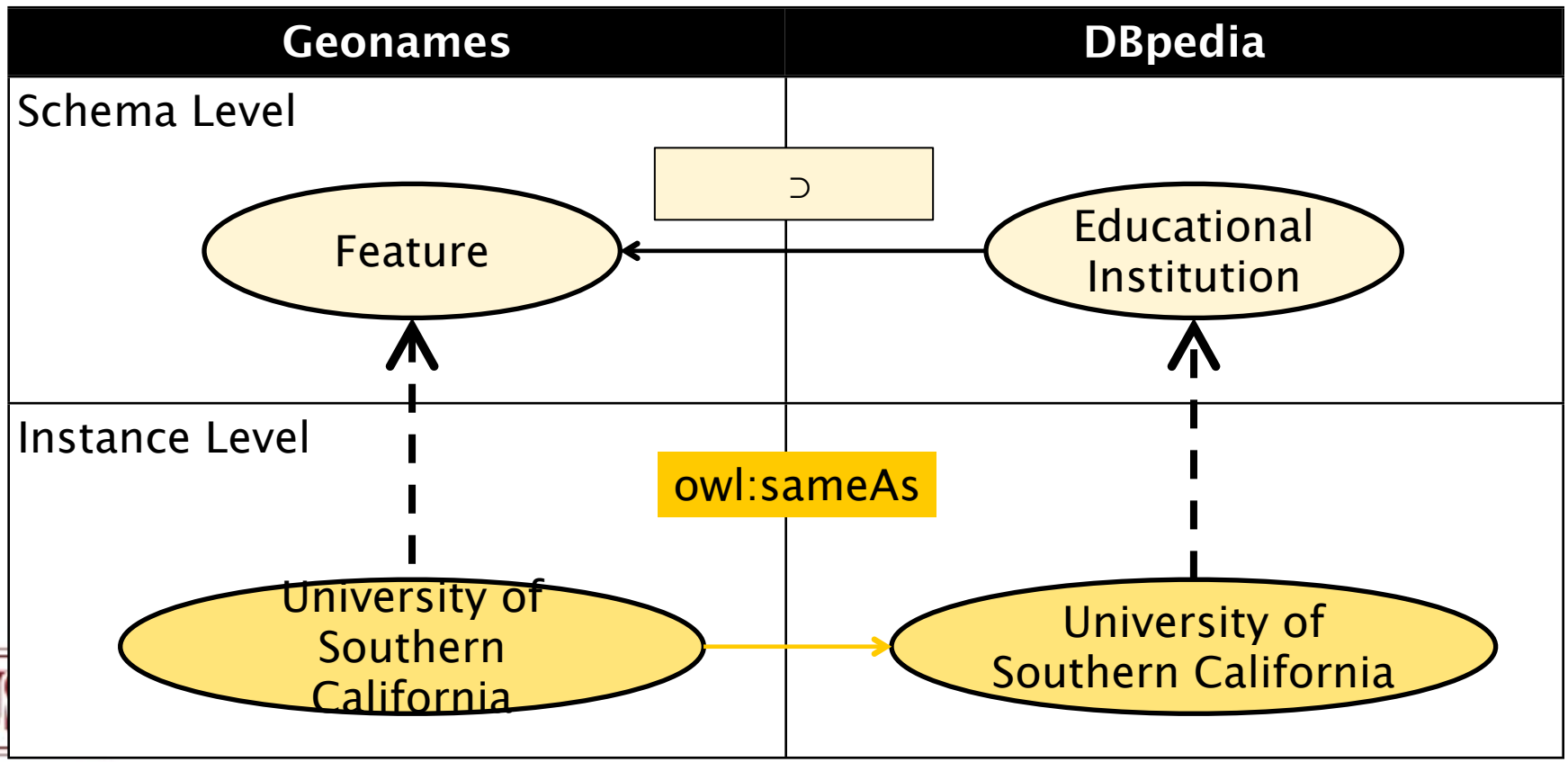
- Ontologies can be highly specialized
  - e.g. DBpedia has classes for Educational Institutions, Bridges, Airports, etc.
- Ontologies can be rudimentary
  - e.g. in Geonames all instances only belong to a single class – 'Feature'
  - Derived from RDBMS schemas from which Linked Data was generated
- There might not exist exact equivalences between classes in two sources

USC

USC Viterbi School of Engineering

Traditional Alignments

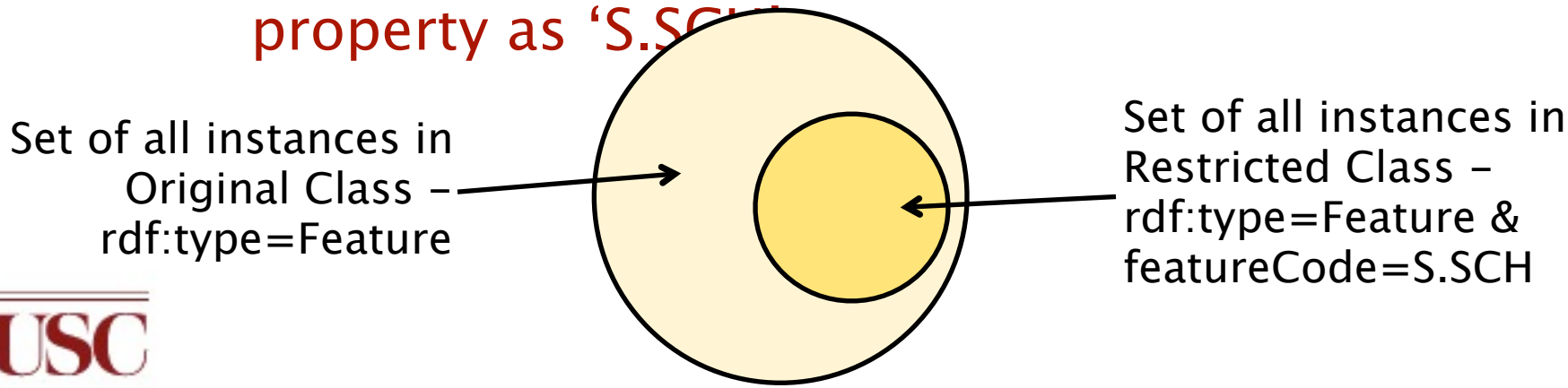
- Only subset relations possible with difference in class specializations



USC Viterbi School of Engineering

Restriction Classes

- A specialized class can be created by restricting the value of one or more properties
- The following Venn diagram explains a restriction class in Geonames with a restriction on the value of the featureCode property as 'S.SCH'

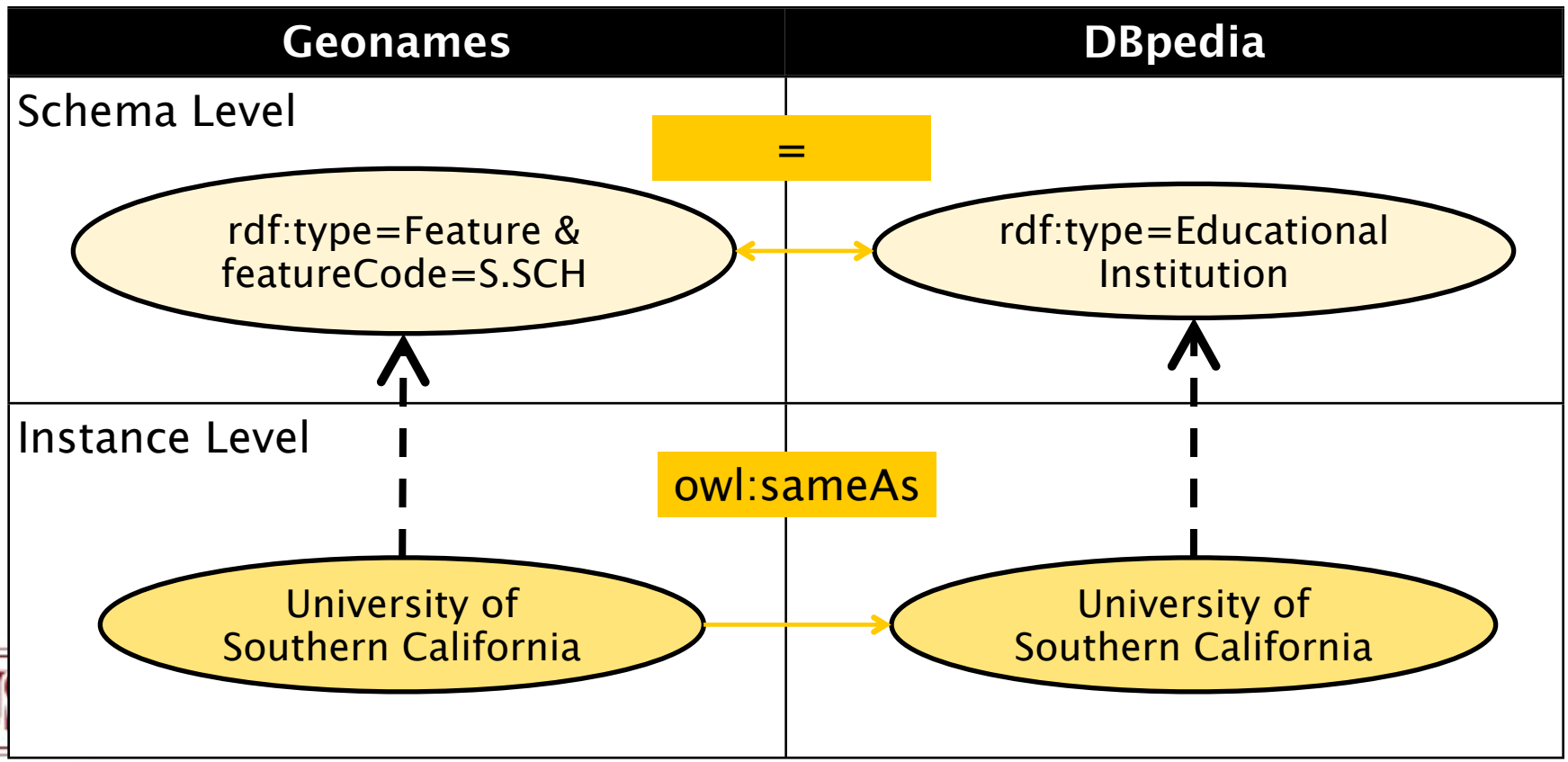


USC

USC Viterbi School of Engineering

Objective 2: Find Alignments Between Restriction Classes

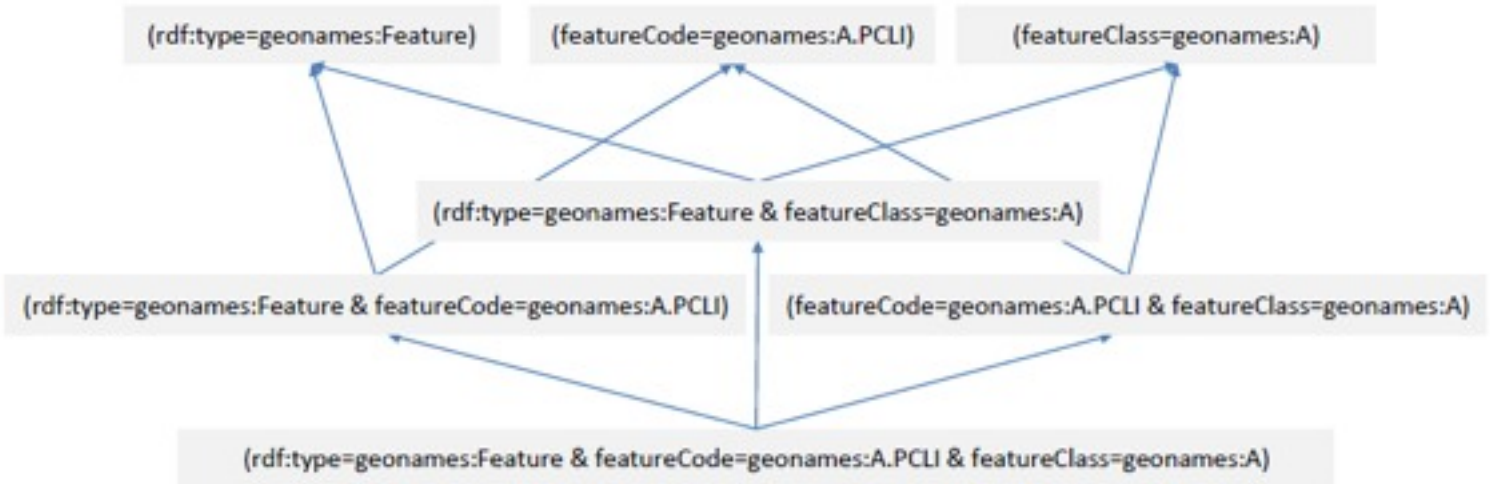
- Find and model specialized descriptions of classes



USC Viterbi School of Engineering

Nature of Restriction Classes

- Instances belonging to a restriction class also belong to parent restriction class
  - e.g. restrictions from Geonames below



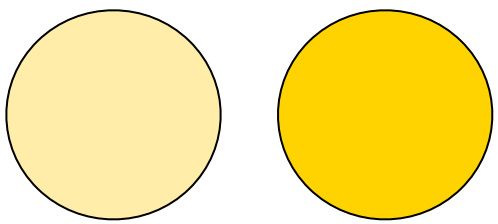
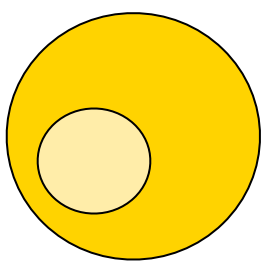
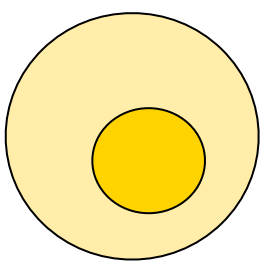
- This also results in a hierarchy in the alignments, which our algorithm exploits

USC

Extensional Approach to Ontology Alignment

Represents set of instances belonging to ClassA

Represents set of instances belonging to ClassB

ClassA is disjoint from ClassB 	ClassA is equivalent to ClassB
ClassA is subset of ClassB 	ClassB is subset of ClassA 

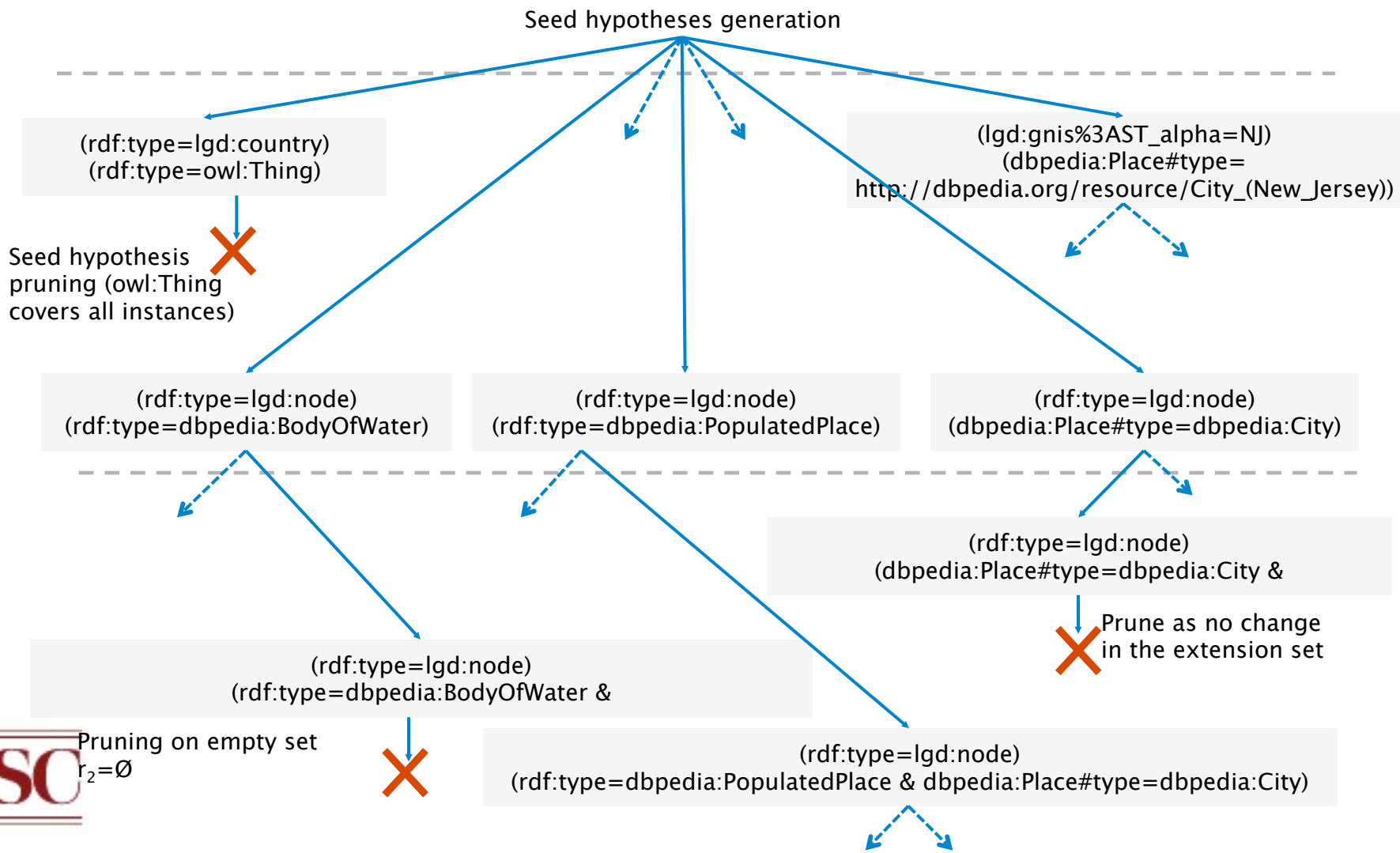
USC Viterbi School of Engineering

Alignment Hypotheses

- An alignment hypothesis considers aligning
  - a restriction class from ontology O_1
  - another restriction class from ontology O_2
- Find relation between the two restriction classes
  - using extensional comparison on set of instances belonging to each restriction class
  - Use instance pair identifiers from pre-processing step (combination of URIs of linked instances)

USC

Exploration of Hypotheses Search Space



USC Viterbi School of Engineering

Example Alignments from LinkedGeoData, Geonames, and DBpedia

#	LinkedGeoData *restriction*	DBpedia *restriction*	Relation
1	rdf:type=lgd:node	rdf:type=owl:Thing	$r_1 = r_2$
2	rdf:type=lgd:aerodrome	rdf:type=dbpedia:Airport	$r_1 = r_2$
3	rdf:type=lgd:island	rdf:type=dbpedia:Island	$r_1 = r_2$
4	lgd:gnis_%3AST_alpha=NJ	dbpedia:Place#type= http://dbpedia.org/resource/City_(New_Jersey)	$r_1 = r_2$
5	rdf:type=lgd:village	rdf:type=dbpedia:PopulatedPlace	$r_1 \subset r_2$
#	Geonames *restriction*	DBpedia *restriction*	Relation
6	geonames:featureClass=geonames:P	rdf:type=dbpedia:PopulatedPlace	$r_1 = r_2$
7	geonames:featureClass=geonames:H	rdf:type=dbpedia:BodyOfWater	$r_1 = r_2$
8	geonames:parentFeature=http://sws.geonames.org/3174618/	dbpedia:City_region=http://dbpedia.org/resource/Lombardy	$r_1 = r_2$
9	geonames:featureCode=geonames:S.SCH	rdf:type=dbpedia:EducationalInstitution	$r_1 = r_2$
10	geonames:featureCode=geonames:S.SCH & geonames:inCountry=geonames:US	rdf:type=dbpedia:EducationalInstitution	$r_1 = r_2$
11	geonames:featureCode=geonames:T.MT	rdf:type=dbpedia:Mountain	$r_1 \subset r_2$

USC

USC Viterbi School of Engineering

Outline of the Talk

- Linked Open Data
  - Building and linking ontologies of linked data
- Linked Open Services
  - Building semantic web services from the Deep Web
- Discussion
  - Remaining challenges

USC

USC Viterbi School of Engineering

Building semantic web services from the Deep Web [Ambite et al., ISWC 2009]

- Automatically build semantic models for data and services available on the larger Web
- Construct models of these sources that are sufficiently rich to support querying and integration
  - Build models for the vast amount of structured and semi-structured data available
    - Not just web services, but also form-based interfaces
    - E.g., Weather forecasts, flight status, stock quotes, currency converters, online stores, etc.
  - Learn models for information-producing web sources and web services

USC

USC Viterbi School of Engineering

Approach

- Start with an some initial knowledge of a domain
  - Sources and semantic descriptions of those sources
- Automatically
  - Discover related sources
  - Determine how to invoke the sources
  - Learn the syntactic structure of the sources
  - Identify the semantic types of the data
  - Build semantic models of the source

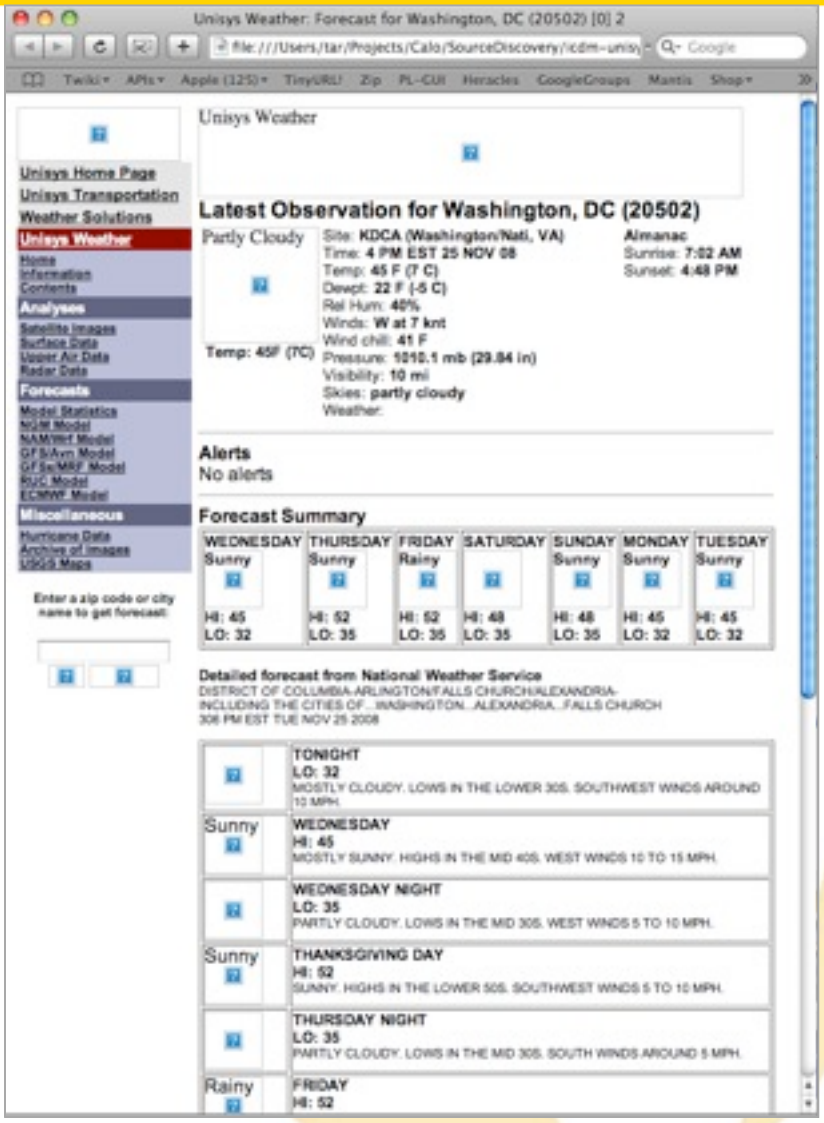
USC

USC Viterbi School of Engineering

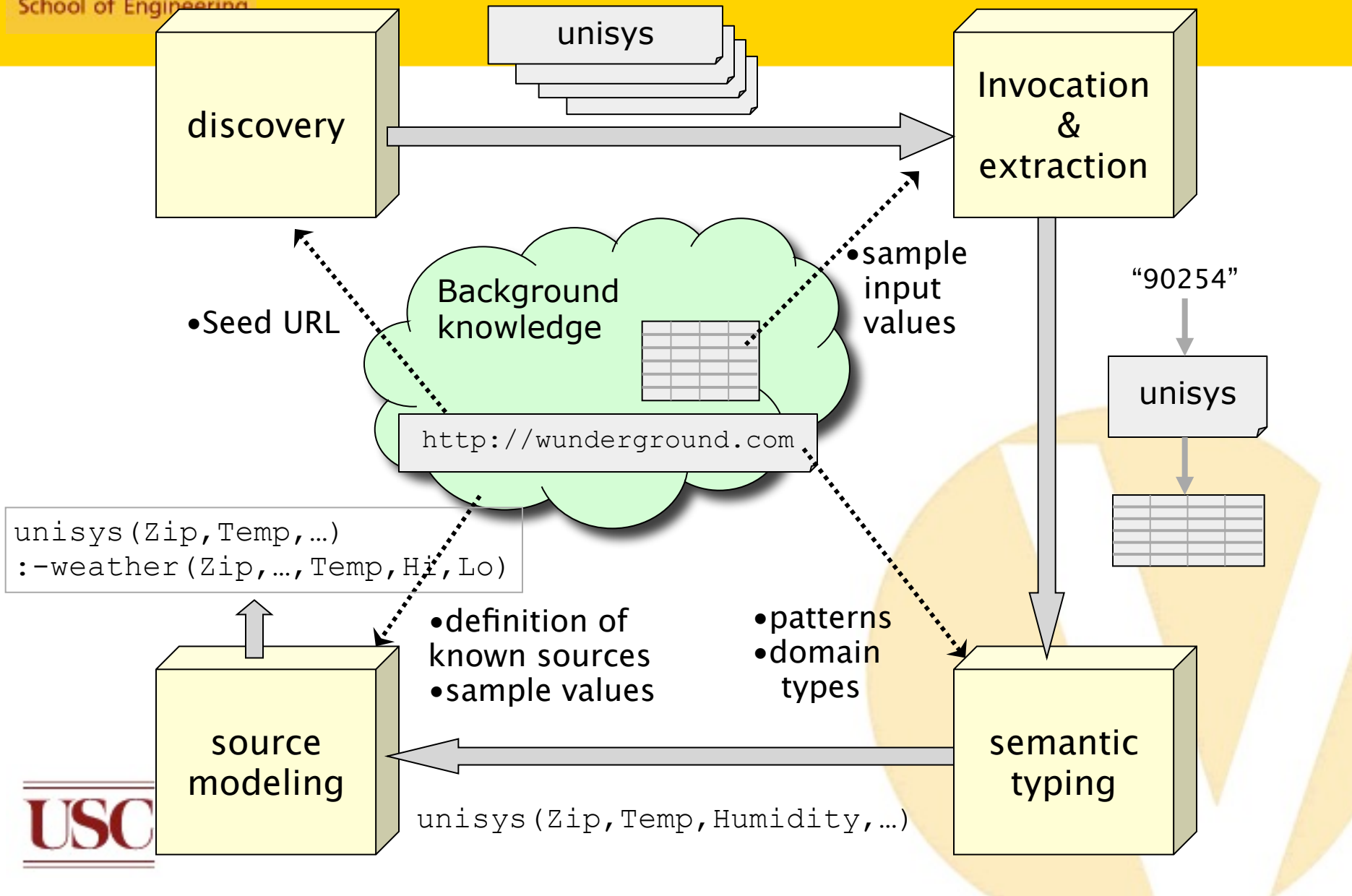
Seed Source

USC Viterbi School of Engineering

Automatically Discover and Build Semantic Web Services for Related Sources



Integrated Approach



Semantic Typing [Lerman, Plangprasopchok, & Knoblock]

✓Idea: Learn a model of the content of data and use it to recognize new examples

Background knowledge

learn

Patterns

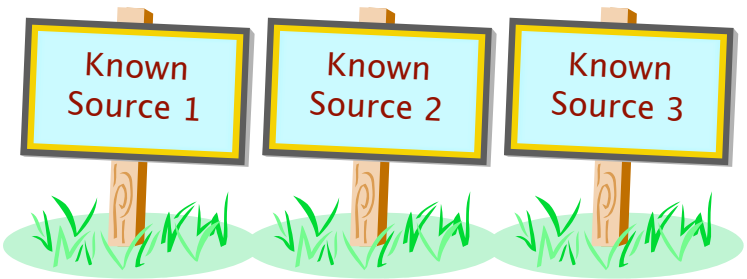
```
:StreetAddress:   :Email:
 4DIG CAPS Rd      ALPHA@ALPHA.edu
 3DIG N CAPS Ave   ALPHA@ALPHA.com
  …                  …
 :State:           :Telephone:
  CA                (3DIG) 3DIG-4DIG
  2UPPER            +1 3DIG 2DIG 4DIG
                    …
```

Person	Address	Work
E Lewis	3518 Hilltop Rd	(419) 531 - 0504
Andrew Lewis	3543 Larchmont Pkwy	(518) 474 -
C. S. Lewis	555 Willow Run Dr	(612) 578 -
Carmen Jones	355 Morgan Ave N	(612) 522 -
John Jones	3574 Brookside Rd	(555) 531 - 9566
Location	**State_prov**	**Postal_code**
Toledo	OH	64325-3000
Toledo	OH	64356
Seattle	WA	8422
Seattle	WA	8435
Omaha	NE	52456-6444

label

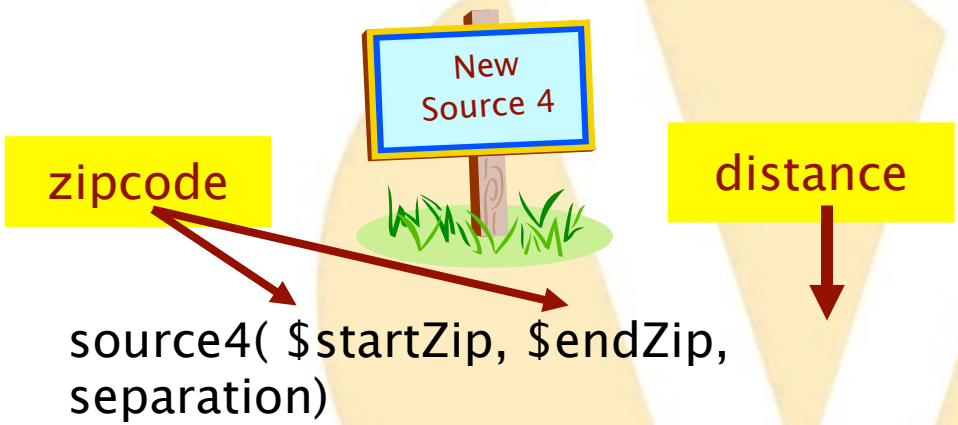
:FullName:	:StreetAddress:	:Telephone:
E Lewis	3518 Hilltop Rd	(419) 531 - 0504
drew Lewis	3543 Larchmont Pkwy	(518) 474 - 4799
. Lewis	555 Willow Run Dr	(612) 578 - 5555
rmen Jones	355 Morgan Ave N	(612) 522 - 5555
John Jones	3574 Brookside Rd	(555) 531 - 9566
:City:	**:State:**	**:Zipcode:**
Toledo	OH	64325-3000
Toledo	OH	64356
Seattle	WA	8422
Seattle	WA	8435
Omaha	NE	52456-6444

Inducing Source Definitions



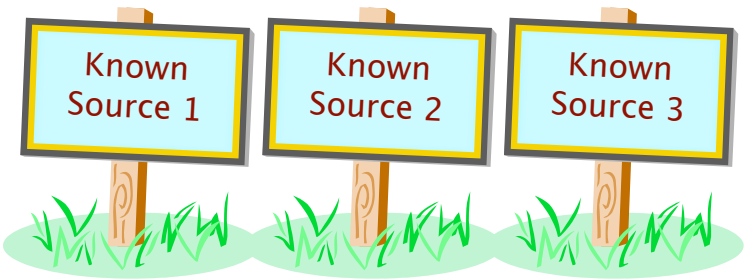

```
source1($zip, lat, long) :-
    centroid(zip, lat, long).
source2($lat1, $long1, $lat2, $long2, dist) :-
    greatCircleDist(lat1, long1, lat2, long2, dist).
source3($dist1, dist2) :-
    convertKm2Mi(dist1, dist2).
```

- Step 1: classify input & output semantic types



Generating Plausible Definition

[Carman & Knoblock, 2007]




```
source1($zip, lat, long) :-
    centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
    greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
    convertKm2Mi(dist1, dist2).
```

- Step 1: classify input & output semantic types
- Step 2: generate plausible definitions

```
source4($zip1, $zip2, dist):-
    source1(zip1, lat1, long1),
    source1(zip2, lat2, long2),
    source2(lat1, long1, lat2, long2, dist2),
    source3(dist2, dist).
```

```
source4($zip1, $zip2, dist):-
    centroid(zip1, lat1, long1),
    centroid(zip2, lat2, long2),
    greatCircleDist(lat1, long1, lat2, long2, dist2),
    convertKm2Mi(dist1, dist2).
```

Invoke and Compare the Definition

- Step 1: classify input & output semantic types
- Step 2: generate plausible definitions
- Step 3: invoke service & compare output

```
source4($zip1, $zip2, dist):-
    source1(zip1, lat1, long1),
    source1(zip2, lat2, long2),
    source2(lat1, long1, lat2, long2, dist2),
    source3(dist2, dist).
```

```
source4($zip1, $zip2, dist):-
   centroid(zip1, lat1, long1),
   centroid(zip2, lat2, long2),
   greatCircleDist(lat1, long1, lat2,
long2,dist2),
   convertKm2Mi(dist1, dist2).
```

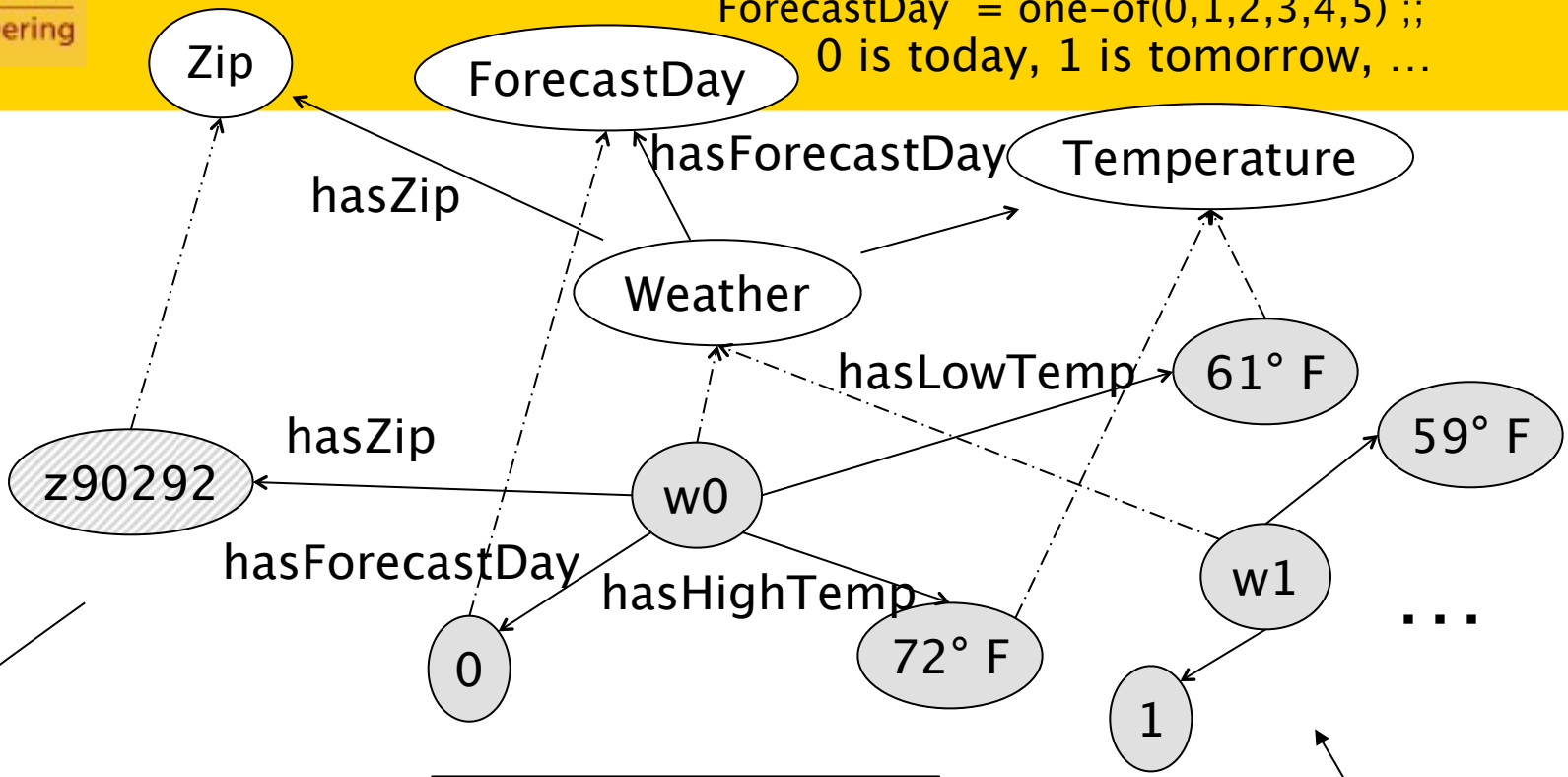
match

$zip1	$zip2	dist (actual)	dist (predicted)
80210	90266	842.37	843.65
60601	15201	410.31	410.83
10005	35555	899.50	899.21

Constructing Semantic Web Services

USC Viterbi School of Engineering

ForecastDay = one-of(0,1,2,3,4,5) ;;
0 is today, 1 is tomorrow, …



z90292 hasName 90292 .

DEIMOS generated Web Service

w1 hasZIP z90292 .
w1 hasTemp 61° F .
…
w1 hasZIP z90292 .
w2 hasLowTemp 59° F .

Legend:

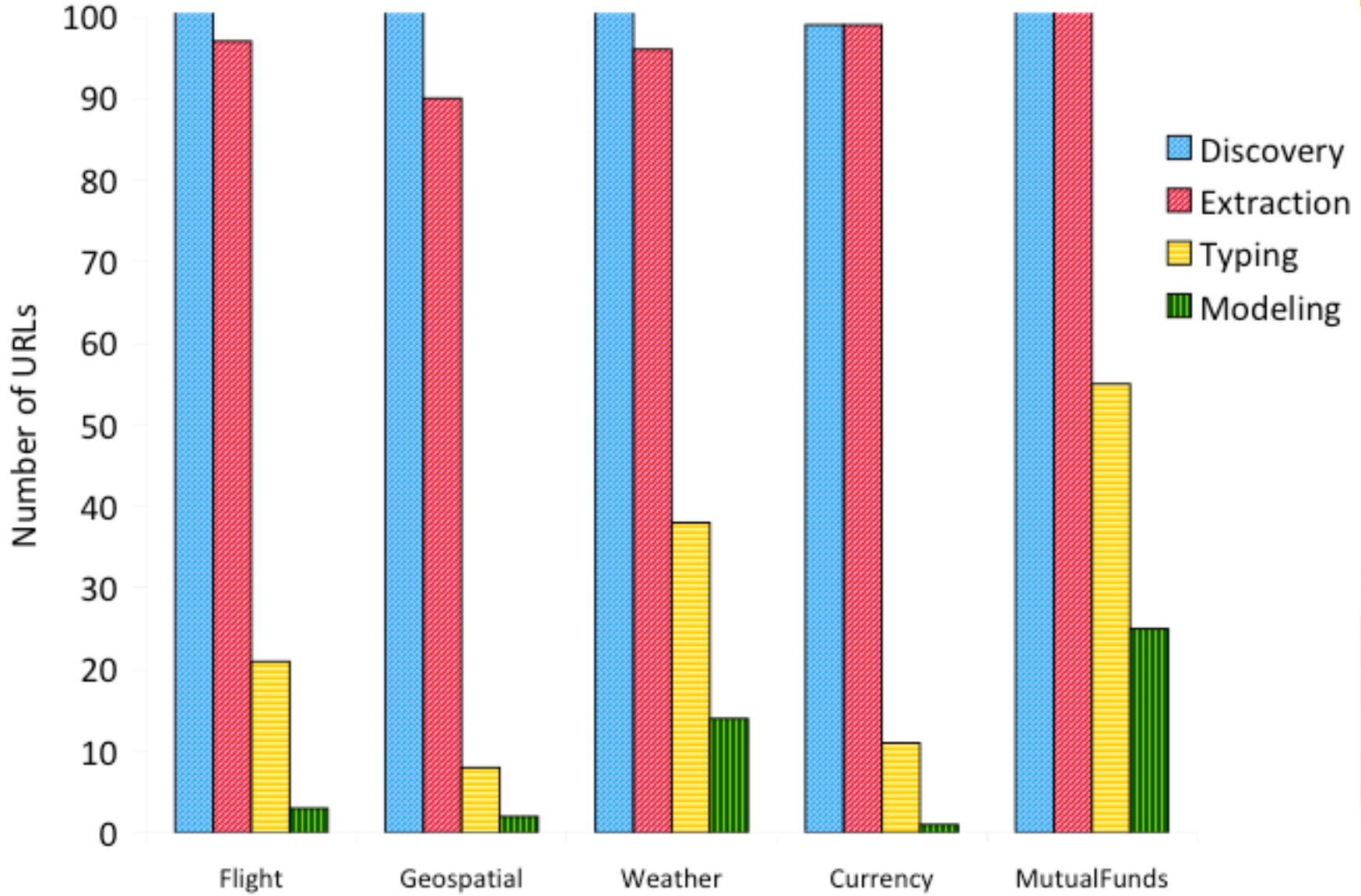
ontology

RDF Input

RDF output

USC Viterbi School of Engineering

Evaluation on Multiple Domains



USC Viterbi School of Engineering

Accuracy of the Models

domain	Precision	Recall	F_1-measure
weather	0.64	0.29	0.39
geospatial	1.00	0.86	0.92
flights	0.69	0.35	0.46
currency	1.00	1.00	1.00
mutualfund	0.72	0.30	0.42

USC

USC Viterbi School of Engineering

Outline of the Talk

- Linked Open Data
  - Building and linking ontologies of linked data
- Linked Open Services
  - Building semantic web services from the Deep Web
- Discussion
  - Remaining challenges

USC

USC Viterbi School of Engineering

Discussion

- Initial work described here just scratches the surface of the problem
  - Goal is to both populate the Web of linked data and have rich semantic models of the data
  - Building semantic descriptions of linked open data will allow us to better understand the available sources and use the sources in a broad range of applications
  - Methods for automatically constructing linked open services will improve the coverage and quality of the sources available

USC

USC Viterbi School of Engineering

Some Challenges

- Linked Open Data
  - How do we build build an overall class hierarchy for a source
  - How do the relations map across sources
  - What do we do about missing and extraneous links
- Linked Open Services
  - How do we improve the accuracy of the learned semantic descriptions
  - How can we learn semantic descriptions that go beyond the current sources
  - How do we learn mappings between enumerated types (e.g., “Arrived” vs. “Landed”)

USC