USC Viterbi
School of Engineering

Aligning Ontologies of Geospatial Linked Data

Rahul Parundekar, Craig A. Knoblock and Jose-Luis Ambite

University of Southern California

USC

USC Viterbi
School of Engineering

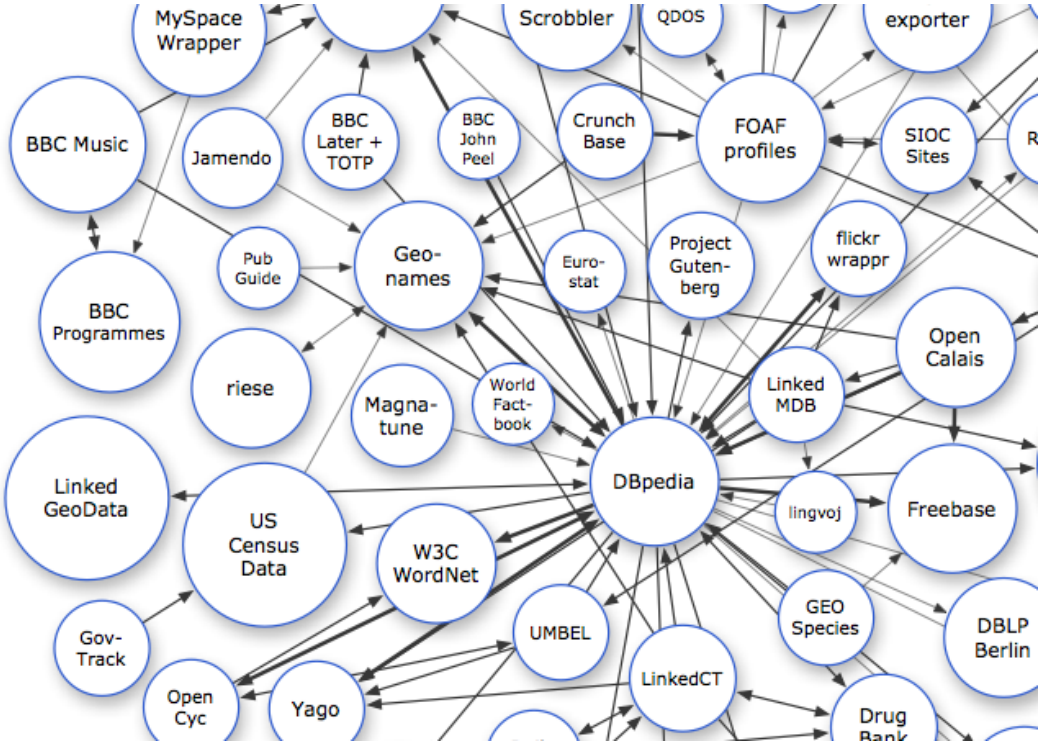
INTRODUCTION

USC

USC Viterbi
School of Engineering

Web of Linked Data

- Vast collection of interlinked information
- Different sources with different schemas



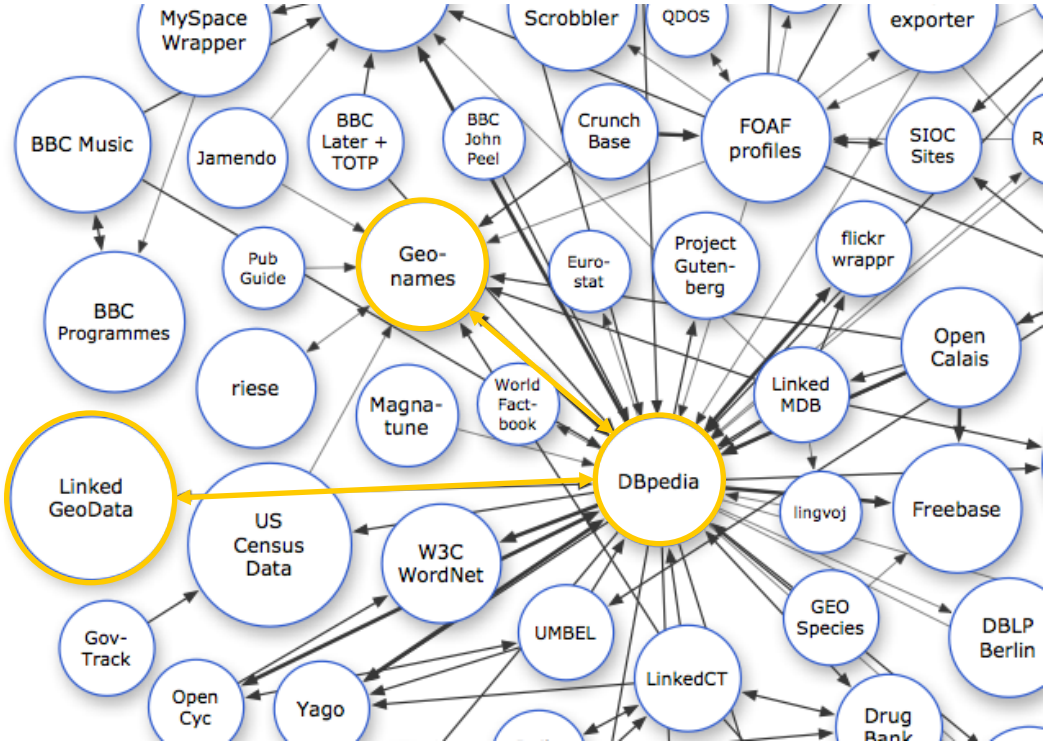
USC

USC Viterbi School of Engineering

Web of Linked Data

- Interlinked instances in the geospatial domain
- Equivalent instances linked with *owl:sameAs*

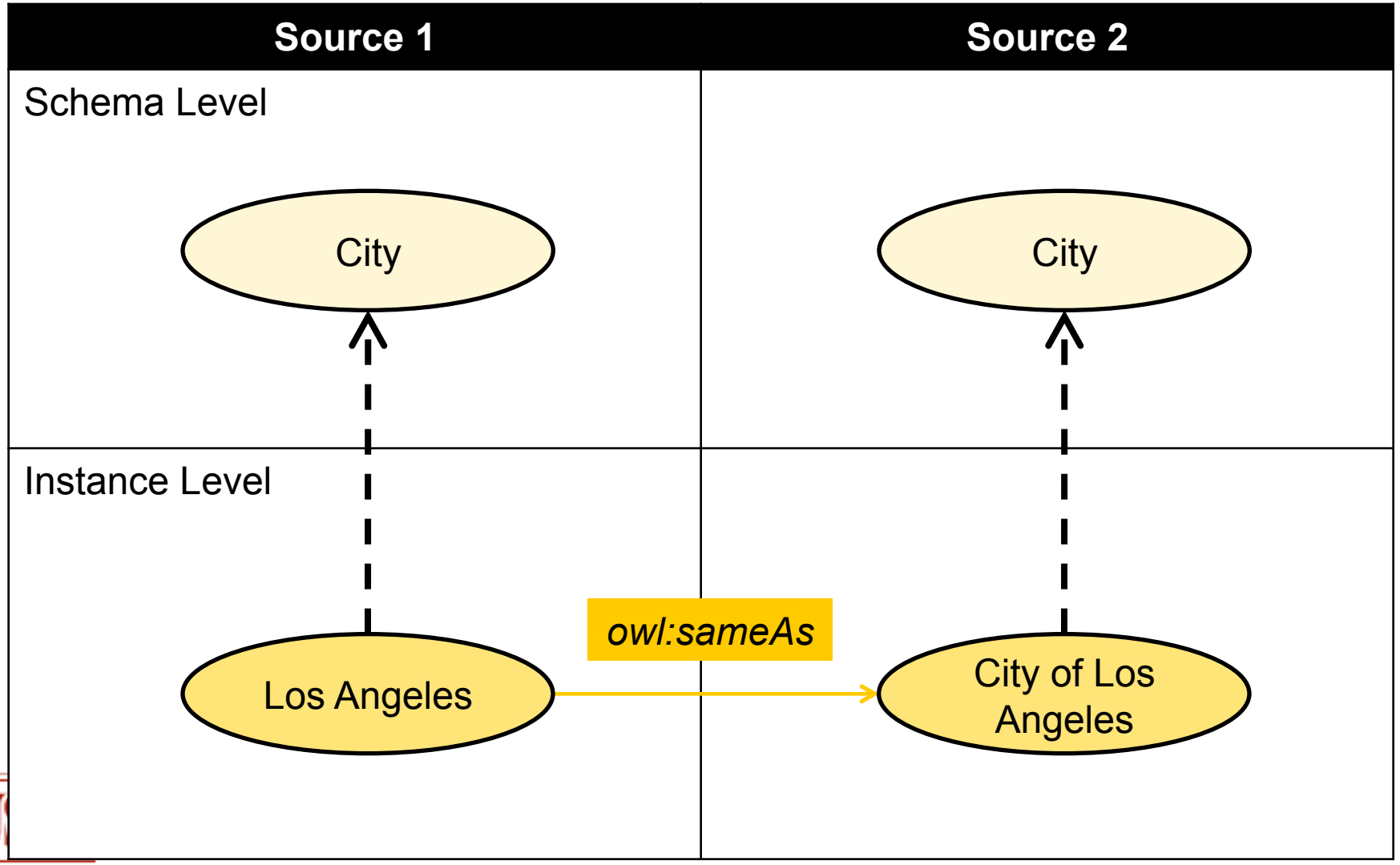
Geospatial Domain



USC

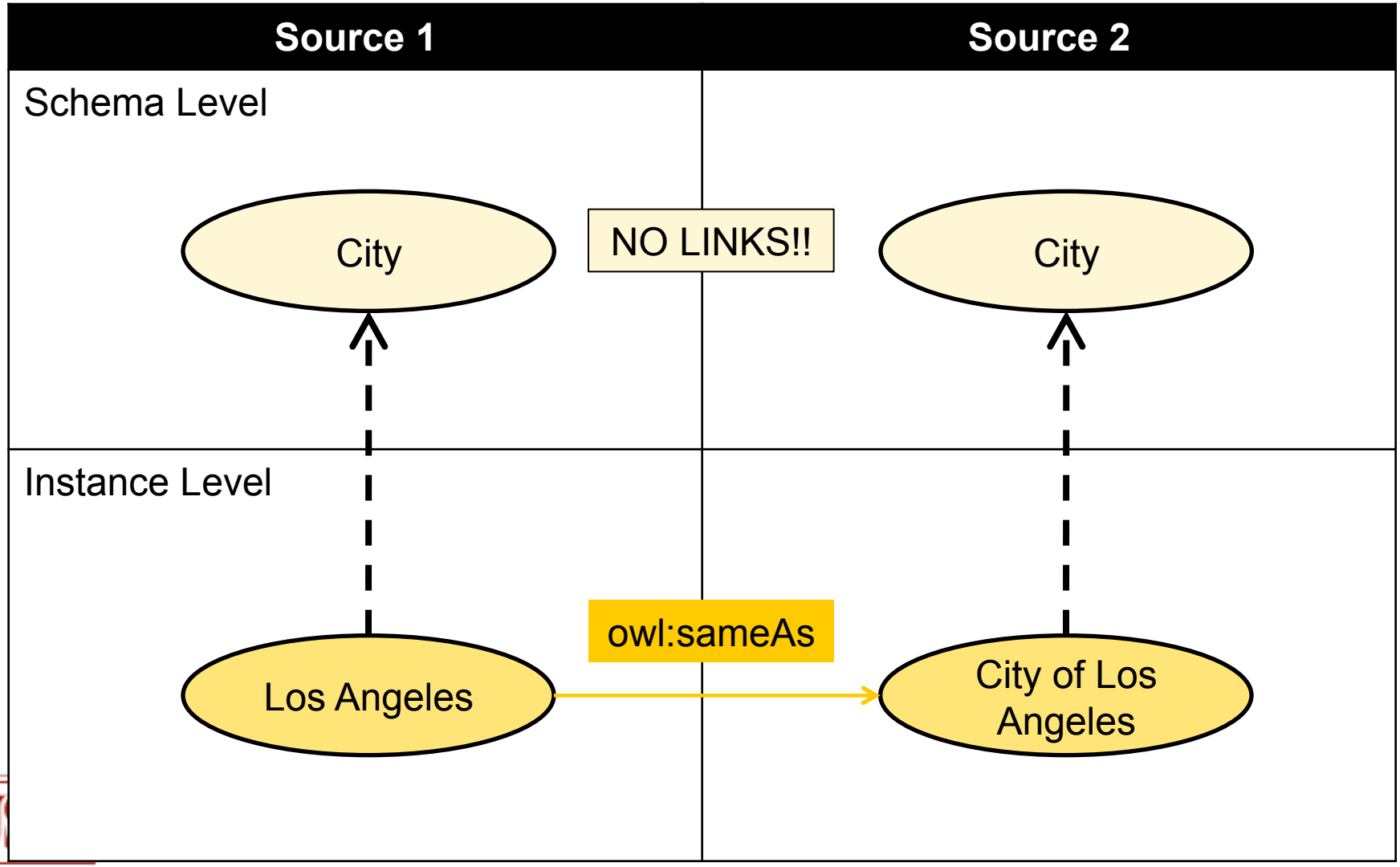
USC Viterbi School of Engineering

Interlinked Instances



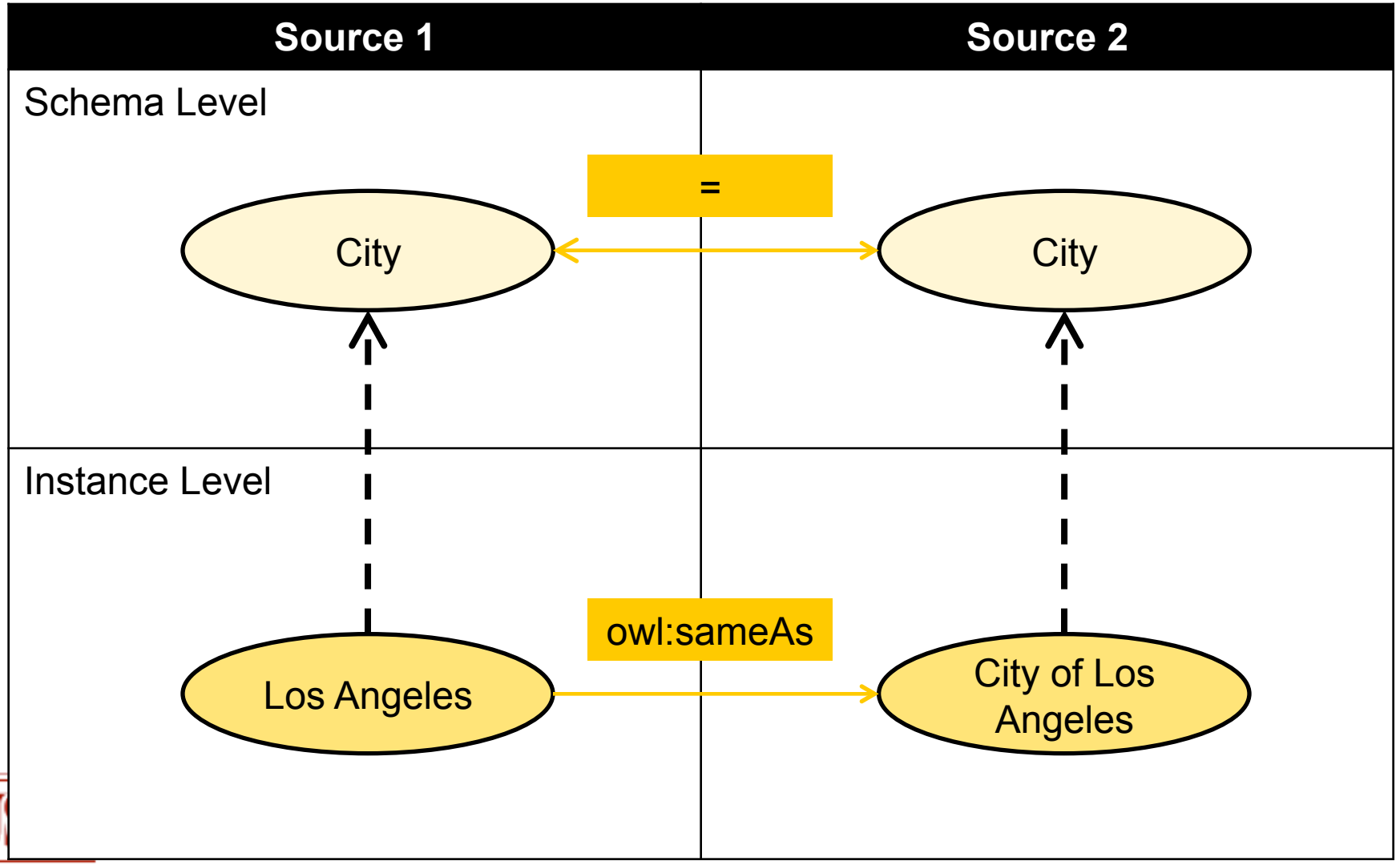
USC Viterbi School of Engineering

Disjoint Schemas



USC Viterbi School of Engineering

Objective 1: Find Schema Alignments



USC Viterbi School of Engineering

Ontologies of Linked Data

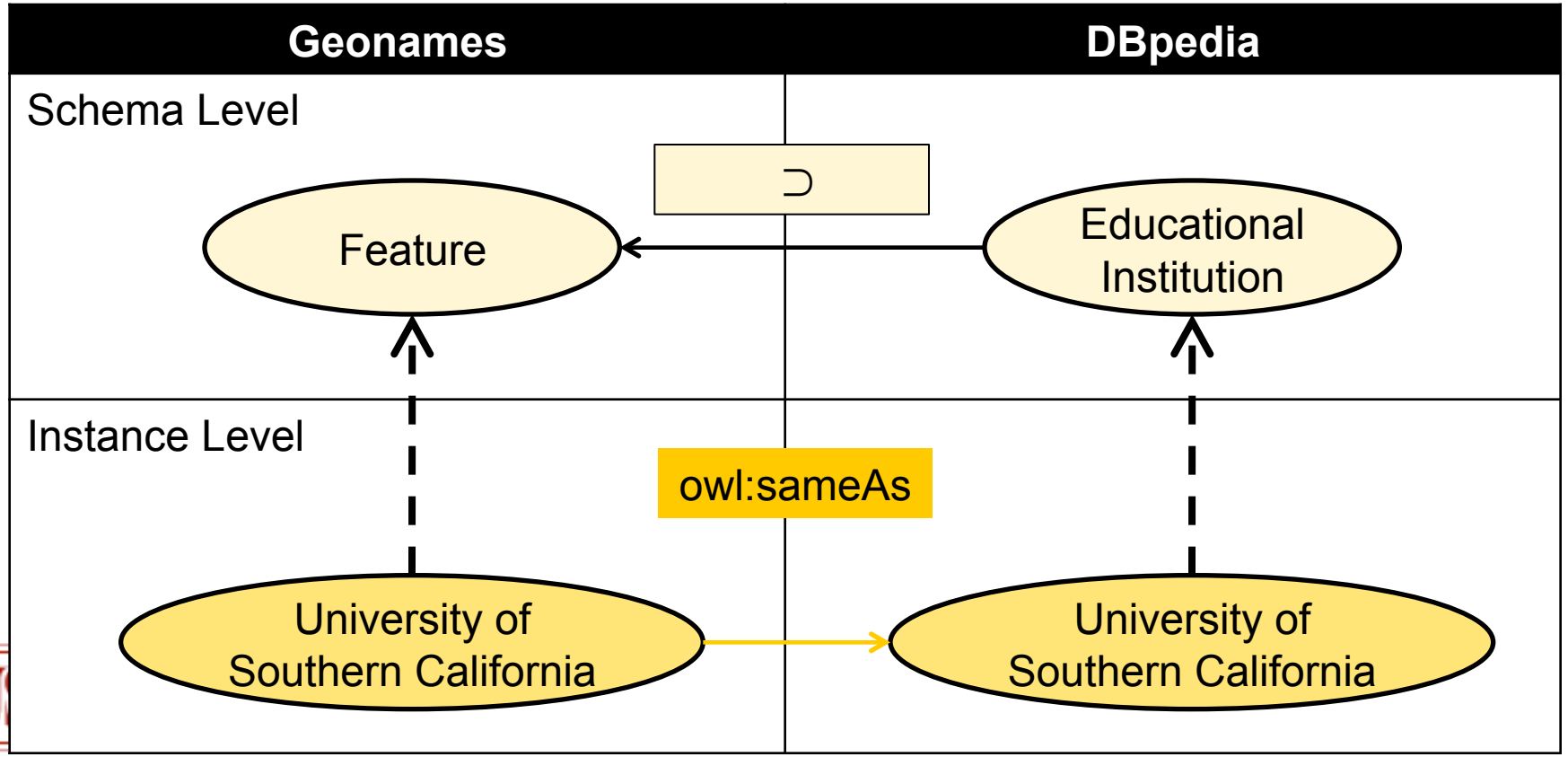
- Ontologies can be highly specialized
  - e.g. DBpedia has classes for *Educational Institutions, Bridges, Airports, etc.*
- Ontologies can be rudimentary
  - e.g. in Geonames all instances only belong to a single class – 'Feature'
  - Derived from RDBMS schemas from which Linked Data was generated
- 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

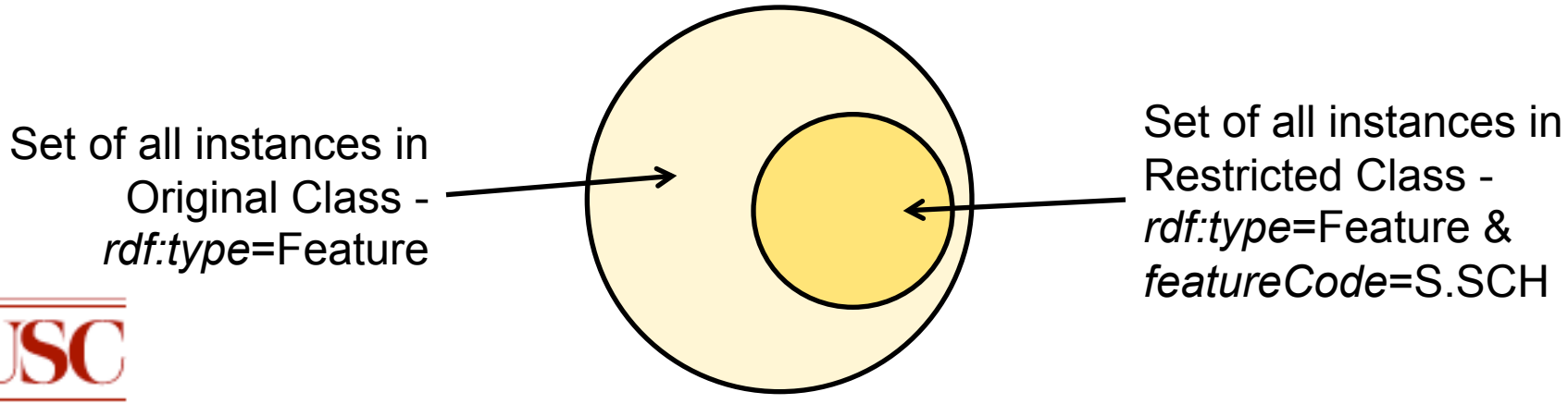


USC Viterbi School of Engineering

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'

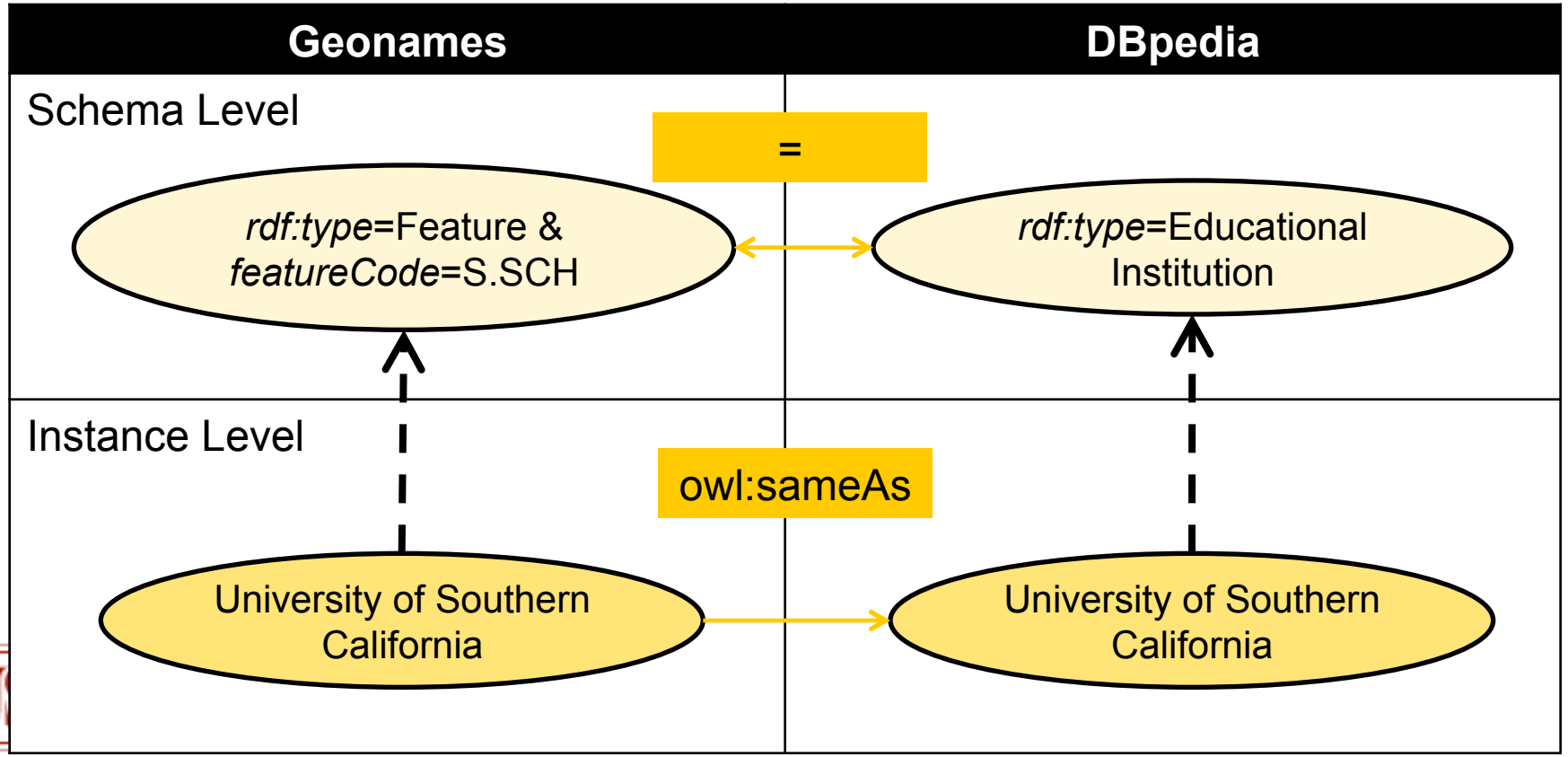


USC

USC Viterbi School of Engineering

Objective 2: Find Alignments Between Restriction Classes

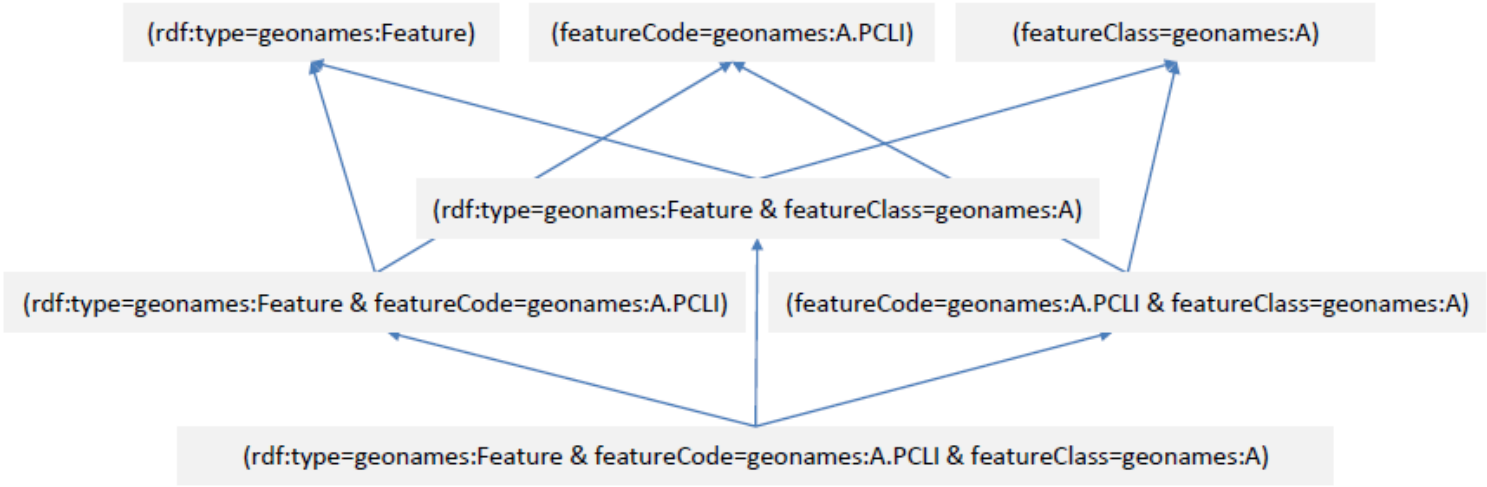
- Find and model specialized descriptions of classes



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

USC Viterbi
School of Engineering

APPROACH

USC

USC Viterbi School of Engineering

Geospatial Data Sources

- Dbpedia
  - 1043 properties 1.5M typed instances
  - Contains Geospatial and other data (e.g. Music, Plants, etc.)
  - Example properties: *Type (City, Peak, Airport)*
- LinkedGeoData
  - 5087 properties 11M instances
  - Contains points of interests like bars, restaurants, etc.
  - Not all instances have a link to DBpedia
- Geonames
  - 17 properties 6.9M instances
  - Example properties: *Type* (Feature), *FeatureClass* (Place, Building, Mountain, etc.), *FeatureCodes* (City, Country, Bridge, Airport, School, etc.)

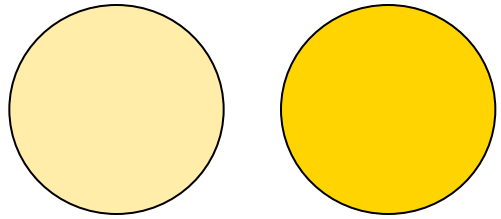
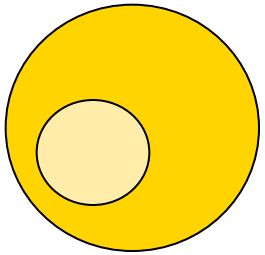
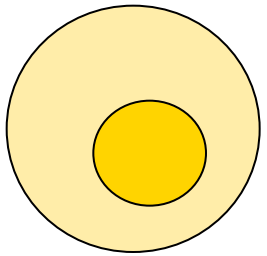
USC

USC Viterbi School of Engineering

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

USC Viterbi School of Engineering

Preprocessing Data Sources

1. Only consider instances that are actually linked
   - Reduced set of instances from one source are linked to instances in other source
   - e.g. Instances of type People, Music Albums, etc. from Dbpedia are removed
   - e.g. Properties like *releaseDate* of Music Albums are also removed

2. Remove inverse functional properties (IFP)
   - IFPs uniquely identify instances & hence restriction on them is a singleton
   - e.g. *wikipediaArticle* property in DBpedia points to same article in different languages

USC

USC Viterbi School of Engineering

Preprocessing Data Sources

3. Convert properties & values for each instance into *vectors*
    1. Each vector is a tuple of property-value pairs for one instance
    2. Multi-valued properties result in multiple tuples with same identifier (URI)
4. Perform a join on the equivalence property to create *instance pairs*
    1. Join vectors from both sources based on equivalence property (e.g. *owl:sameAs*)
    2. Each instance pair identified by combination of the instance URIs

USC

USC Viterbi School of Engineering

Alignment Hypotheses

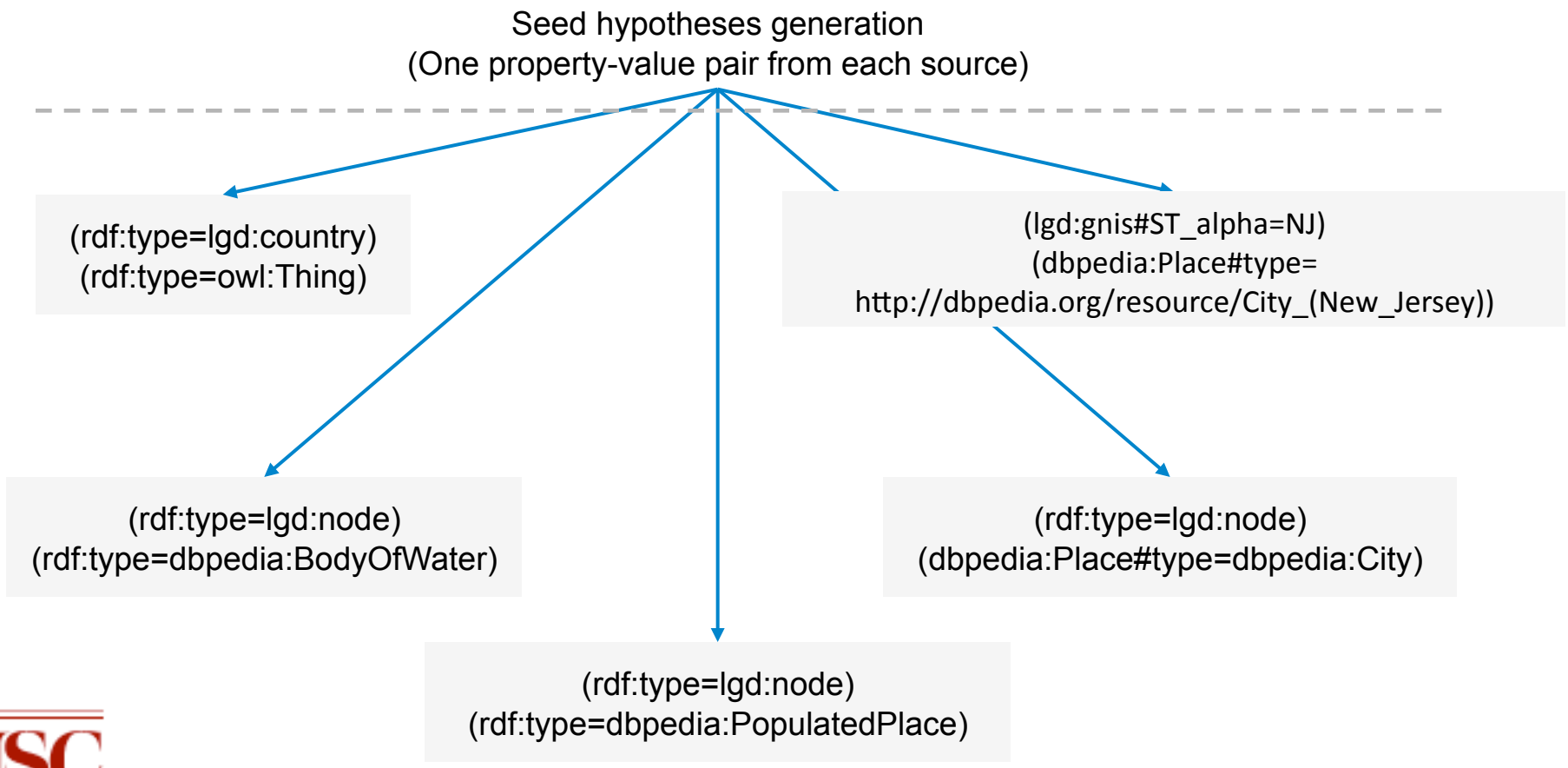
- An alignment hypothesis considers aligning
  - a restriction class from ontology O_1
  - another restriction class from ontology O_2
- Find relation between the two restriction classes
  - using extensional comparison on set of instances belonging to each restriction class
  - Use instance pair identifiers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

USC

USC Viterbi School of Engineering

Top Down Alignment Hypotheses Generation

Aligning LinkedGeoData with DBpedia



USC

USC Viterbi School of Engineering

Exploration of Hypotheses Search Space

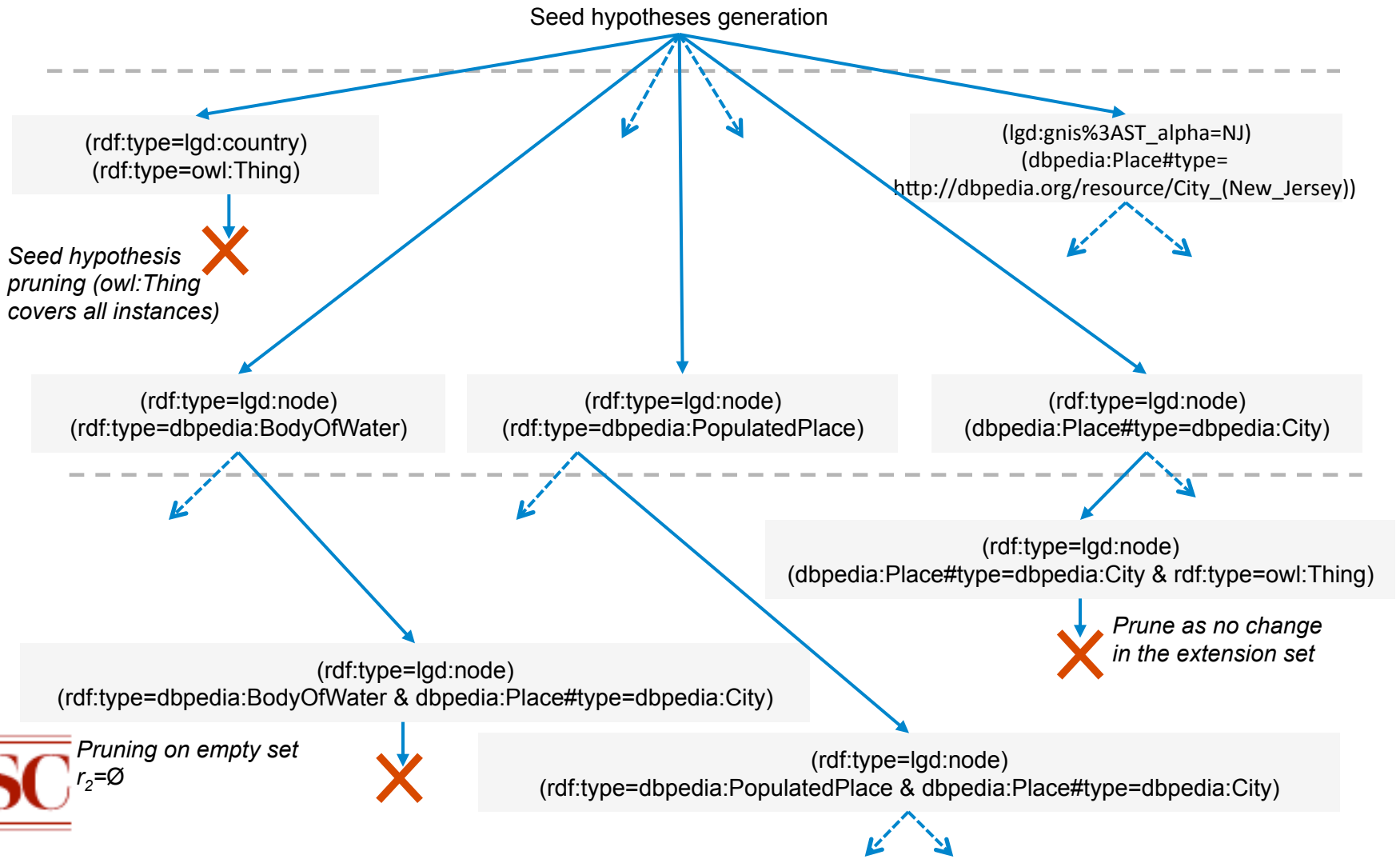
Algorithm:

1. Select a property from O_1
   a. Select one value for the property
   b. Add property-value constraint to restriction from O_1
2. Retain instances belonging to new restriction class
3. Score new alignment and explore its children
4. Repeat steps 1 thru 3 for restriction from O_2
5. Repeat steps 1 thru 4 for all properties

USC

USC Viterbi School of Engineering

Exploration of Hypotheses Search Space



USC Viterbi School of Engineering

Pruning of the Search Space

As the search space is combinatorial we perform several pruning optimizations

1. Number of instance pairs supporting hypothesis must be above a threshold (10 instance pairs)
   - e.g. No City is of type Body of Water

```
(rdf:type=lgd:node)
(rdf:type=dbpedia:BodyOfWater & dbpedia:Place#type=dbpedia:City)
```

2. Prune seed hypothesis if either restriction covers all instances in that source
   - e.g. constraint '*rdf:type*=owl:Thing' covers all instances

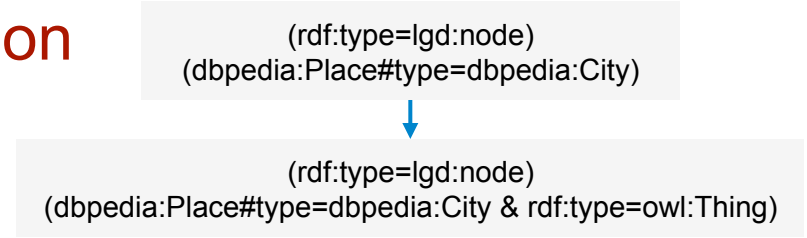
```
(rdf:type=lgd:country)
(rdf:type=owl:Thing)
```

USC

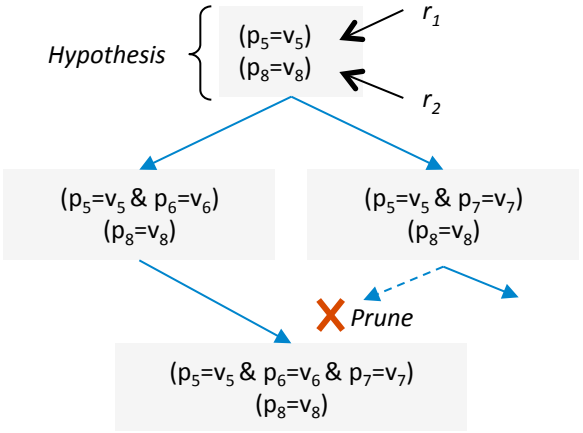
USC Viterbi School of Engineering

Pruning of the Search Space

3. Prune if the added constraint does not change the extension



4. Lexicographic ordering provides a systematic search by pruning hypotheses with reverse order

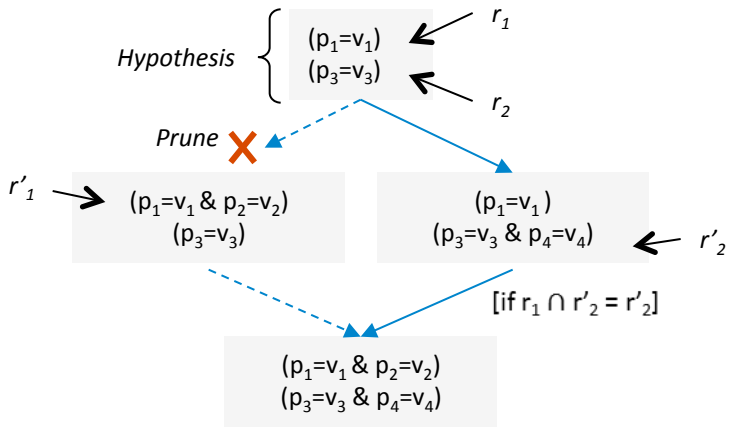


USC

USC Viterbi School of Engineering

Pruning of the Search Space

5. Pruning when $r_1 \cap r_2 = r_1$ (where r_2 is larger than r_1)
   - Any constraint on r'_1 can be explored via other paths



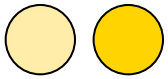
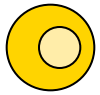
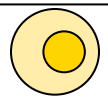
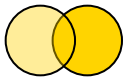
(a) Pruning when $r_1 \cap r_2 = r_1$

USC

USC Viterbi School of Engineering

Relaxed Scoring : Lenient Evaluation

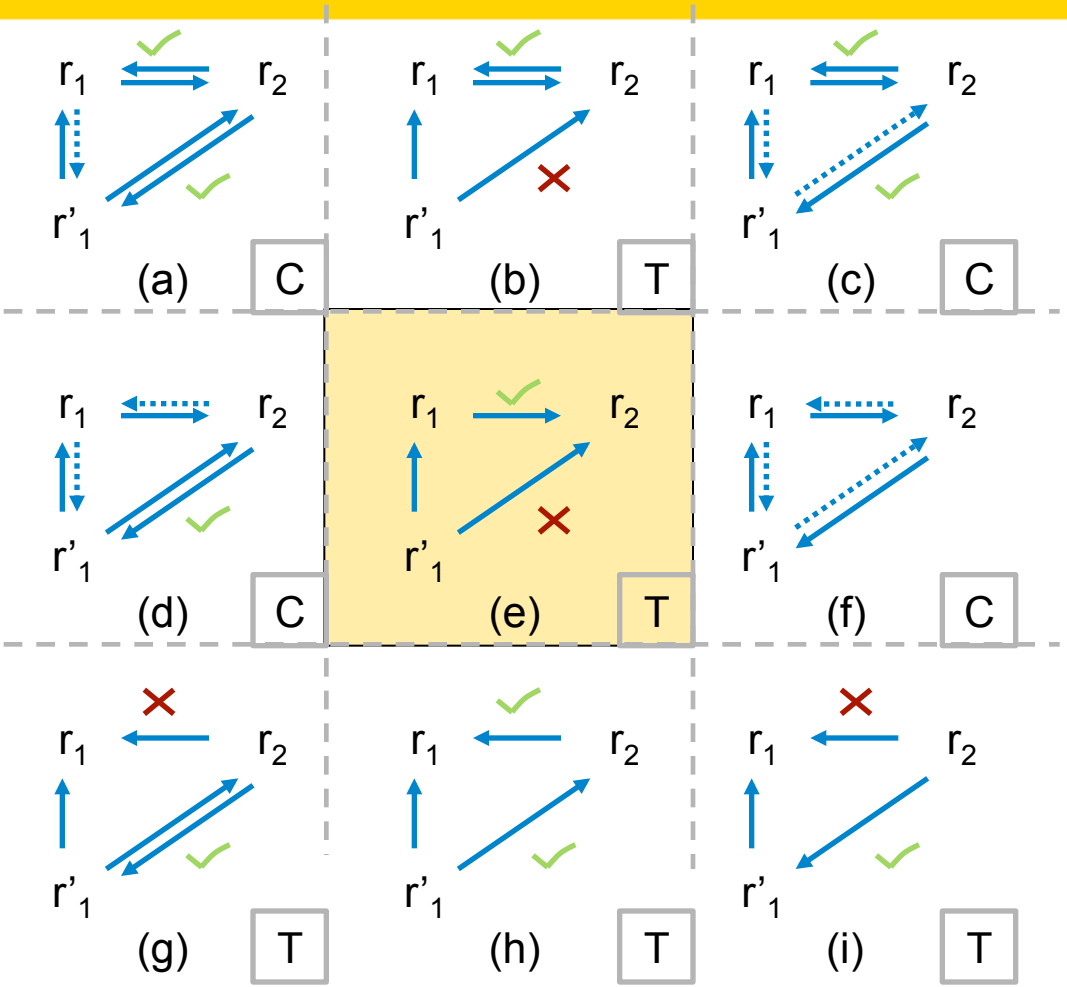
- Compensates for inconsistencies in the data

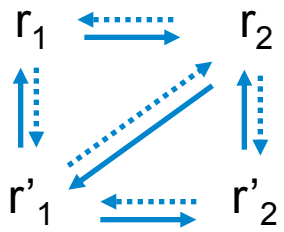
Set Representation	Relation	P = $\frac{\lvert I(r_1) \cap r_2 \rvert}{\lvert r_2 \rvert}$	R = $\frac{\lvert I(r_1) \cap r_2 \rvert}{\lvert r_1 \rvert}$	P'	R'
	Disjoint	= 0	= 0	≤ 0.01	≤ 0.01
	$r_1 \subset r_2$	< 1	= 1	> 0.01	≥ 0.90
	$r_2 \subset r_1$	= 1	< 1	≥ 0.90	> 0.01
	$r_1 = r_2$	= 1	= 1	≥ 0.90	≥ 0.90
	Not enough support	0 < P < 1	0 < R < 1	0.01 < P' < 0.90	0.01 < R' < 0.90

USC

USC Viterbi School of Engineering

Removing Implied Alignments





Cascading

Key:

$r_i \longrightarrow r_j$: Subset relations – $r_i \subset r_j$ found by the algorithm.

$r_i \dashrightarrow r_j$: Implied subset relations.

C : Cycle in subset relations. Hence, all three classes are equivalent.

T : Transitivity in subset relations. One relation can be eliminated.

✗ : Relation eliminated by the T rule.

✓ : Relation retained by the T rule.

USC

USC Viterbi School of Engineering

Results: Preprocessing

- Before Preprocessing

Source	# properties	# instances
LinkedGeoData	5087	11236351
DBpedia	1043	1481003
Geonames	17	6903322

- After Preprocessing

Source 1	# properties after elimination	# instances after reduction	Source 2	# properties after elimination	# instances after reduction	# *vector* combinations	# distinct *instance pairs*
LinkedGeoData	63	23594	DBpedia	16	23632	329641	23632
Geonames	5	71114	DBpedia	26	71317	459716	71317

USC

USC Viterbi School of Engineering

Results: Alignments Found

- Equivalences, Subset alignments before and after removing implied alignments

Source 1 (O_1)	Source 2 (O_2)	$\#(r_1 = r_2)$ total	$\#(r_1 = r_2)$ best matches	$\#(r_1 \subset r_2)$ before	$\#(r_1 \subset r_2)$ after	$\#(r_2 \subset r_1)$ before	$\#(r_2 \subset r_1)$ after
LinkedGeoData	DBpedia	158	152	2528	1837	1804	1627
Geonames	DBpedia	31	19	809	400	1384	1247

USC

USC Viterbi School of Engineering

Geonames Alignment with 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

Conclusion

- Our algorithm generates alignments, consisting of conjunctions of restriction classes
  - Extensional approach on Linked Data
  - Use of restriction classes
- Alignments based on the actual data
  - Implicit closed world assumption means that we determine the relationships based on the data
  - Schemas of linked sources can be readily modeled and used
- Algorithm also able to
  - Specialize ontologies where original were rudimentary
  - Find complimentary hierarchy across an ontologies

USC