

# PCSG: Pattern-Coverage Snippet Generation for RDF Datasets

ISWC 2021

**Xiaxia Wang**<sup>1</sup>, Gong Cheng<sup>1</sup>, Tengteng Lin<sup>1</sup>, Jing Xu<sup>1</sup>, Jeff Z. Pan<sup>2</sup>, Evgeny Kharlamov<sup>3,4</sup>, Yuzhong Qu<sup>1</sup>

<sup>1</sup> State Key Laboratory for Novel Software Technology, Nanjing University, China

<sup>2</sup> School of Informatics, University of Edinburgh, UK

<sup>3</sup> Bosch Center for Artificial Intelligence, Robert Bosch GmbH, Germany

<sup>4</sup> Department of Informatics, University of Oslo, Norway



南京大學  
NANJING UNIVERSITY



THE UNIVERSITY  
of EDINBURGH



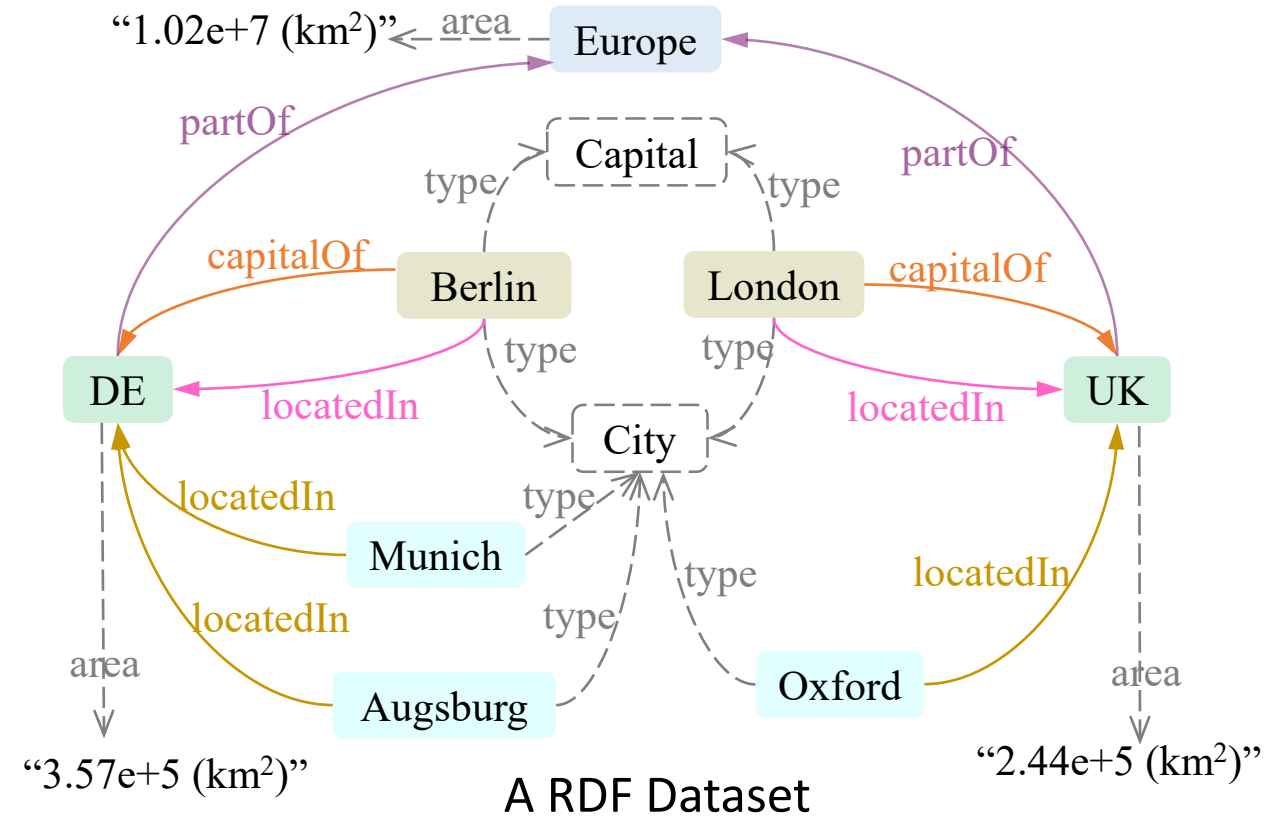
**BOSCH**



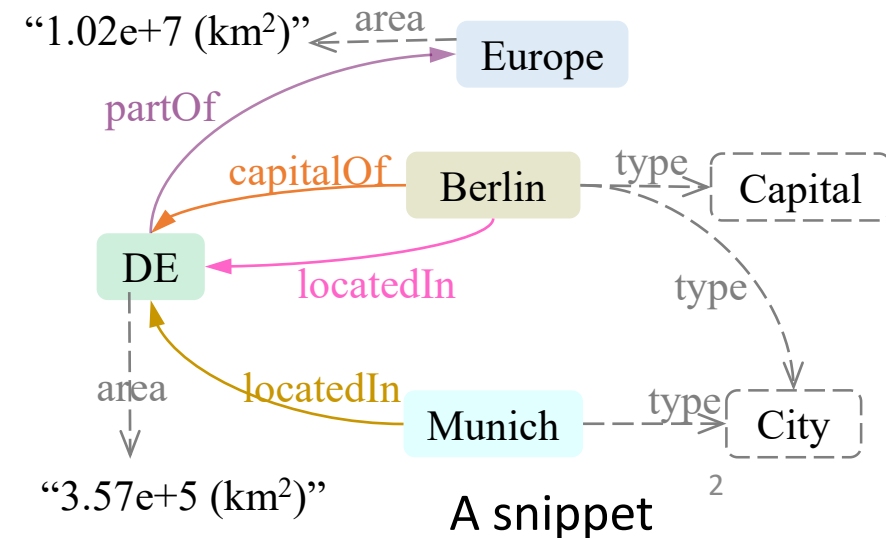
# Background

To ease the comprehension of large complex RDF structure:

- RDF summary: representative schema-level **elements** or **patterns**
- **RDF snippet**: a **connected subgraph** containing frequent classes and properties



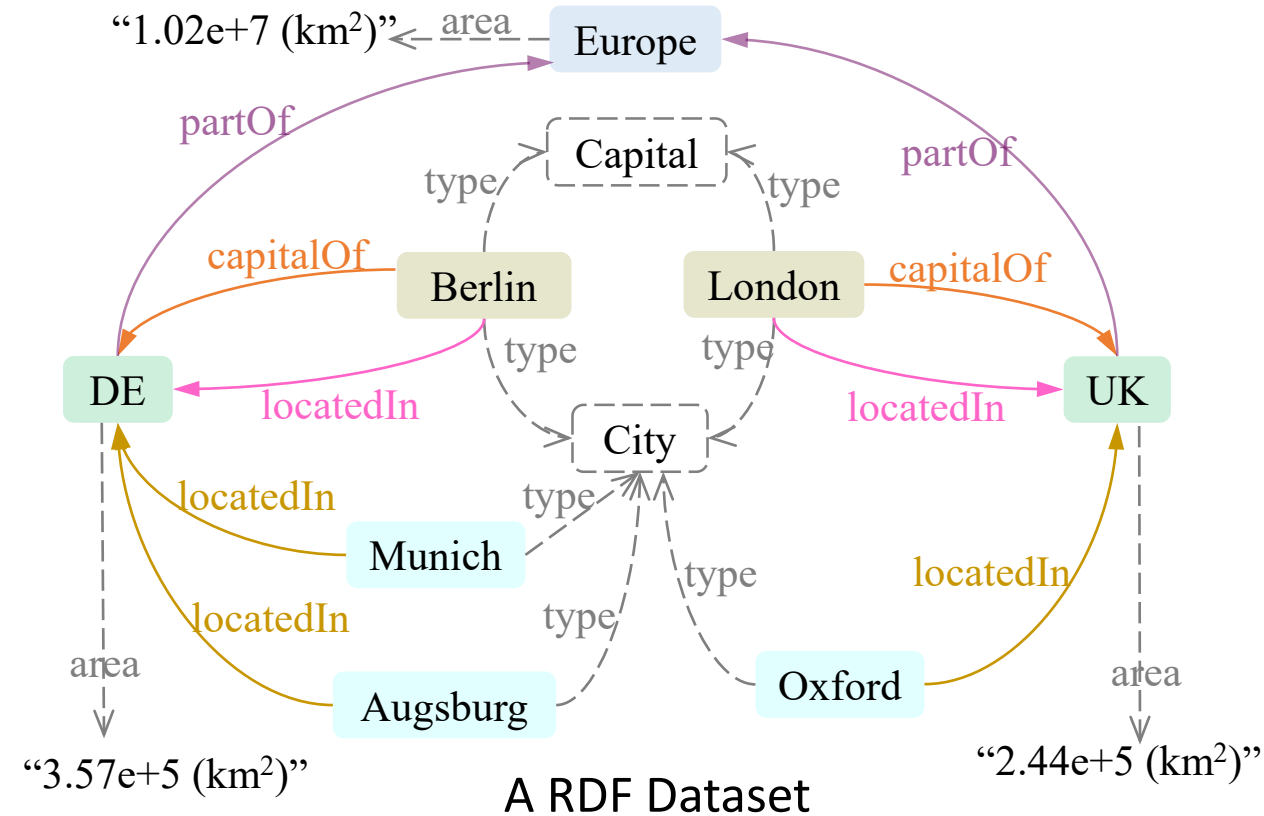
Generation



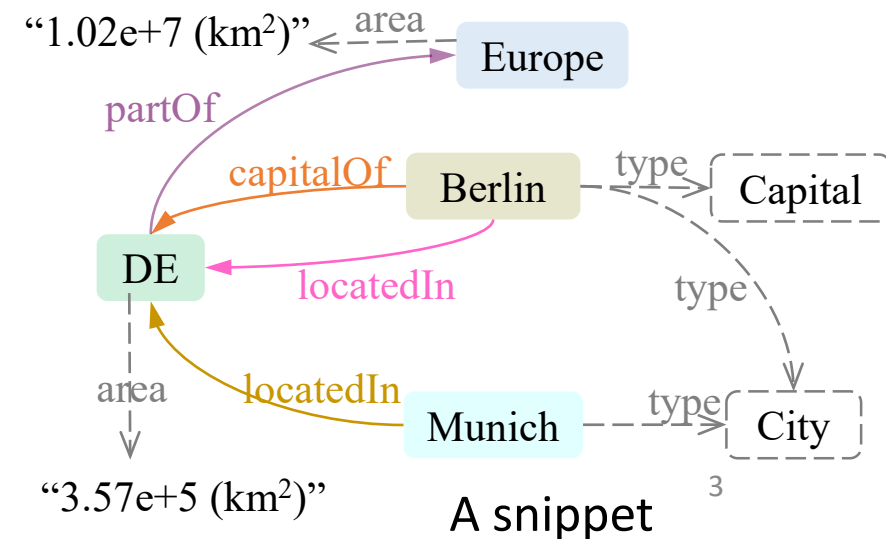
# Motivation and Contribution

Research Questions:

1. How to generate **pattern-coverage** snippets?
2. How to jointly consider **all connected components**?
3. How to be biased towards **keyword queries**?



Generation



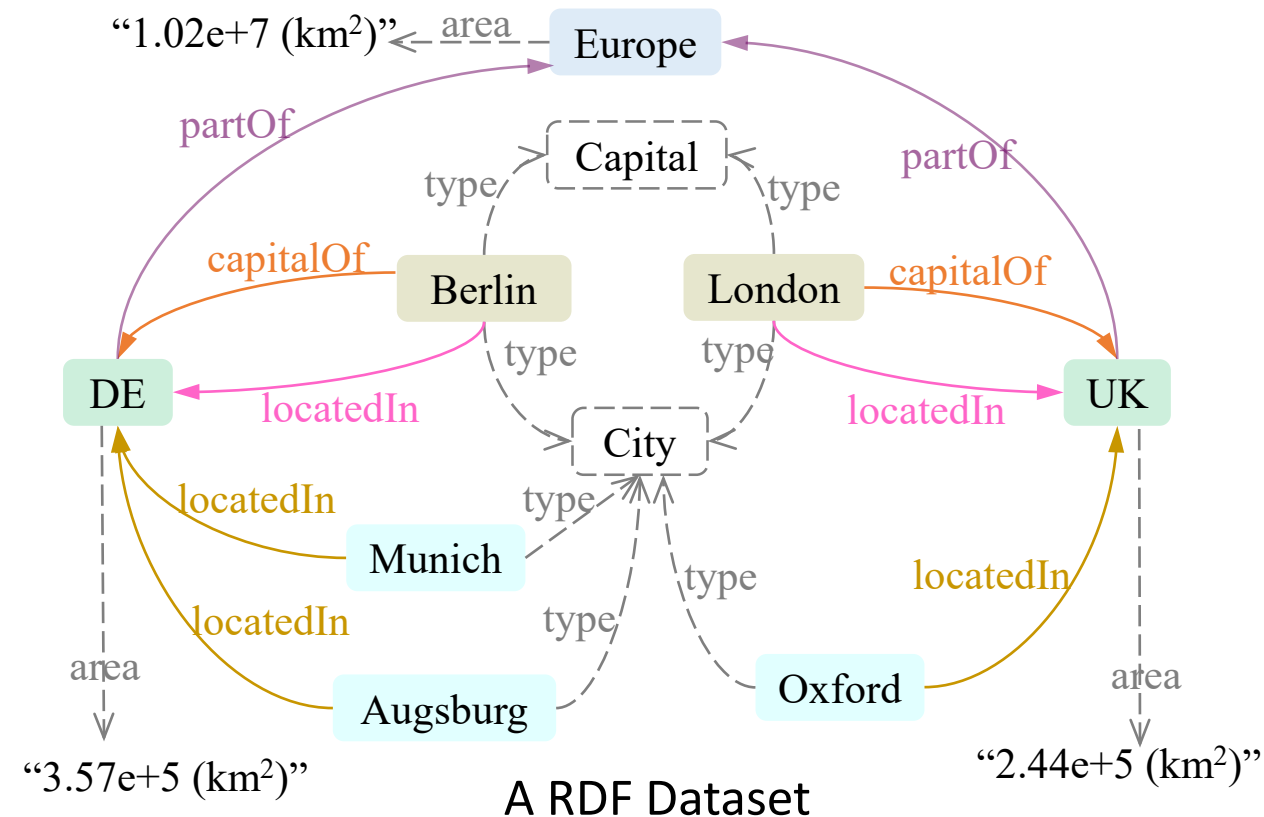
# Motivation and Contribution

Research Questions:

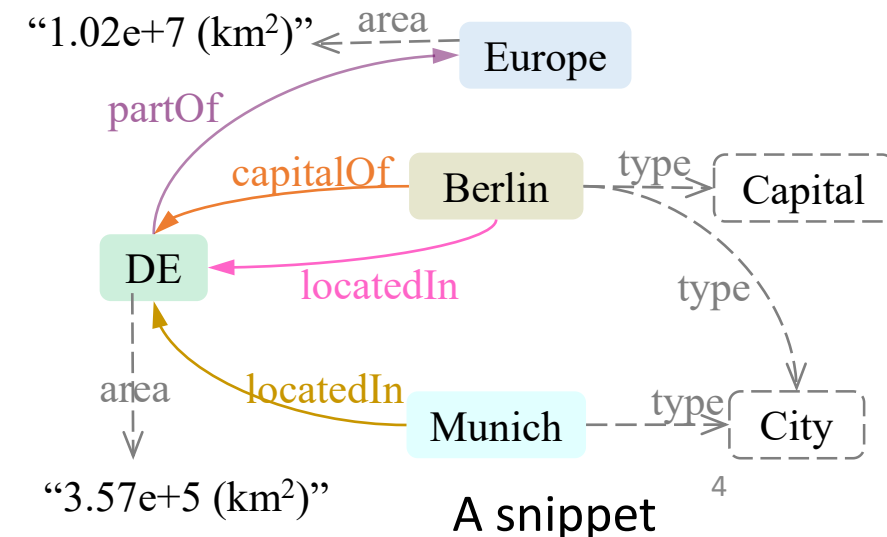
1. How to generate **pattern-coverage** snippets?
2. How to jointly consider **all connected components**?
3. How to be biased towards **keyword queries**?

Contributions:

- Alg. **Basic**: solving a Group Steiner Tree problem
- Alg. **PCSG**: merging **Basic** results among components
- Alg. **QPCSG**: extending **PCSG** to keywords

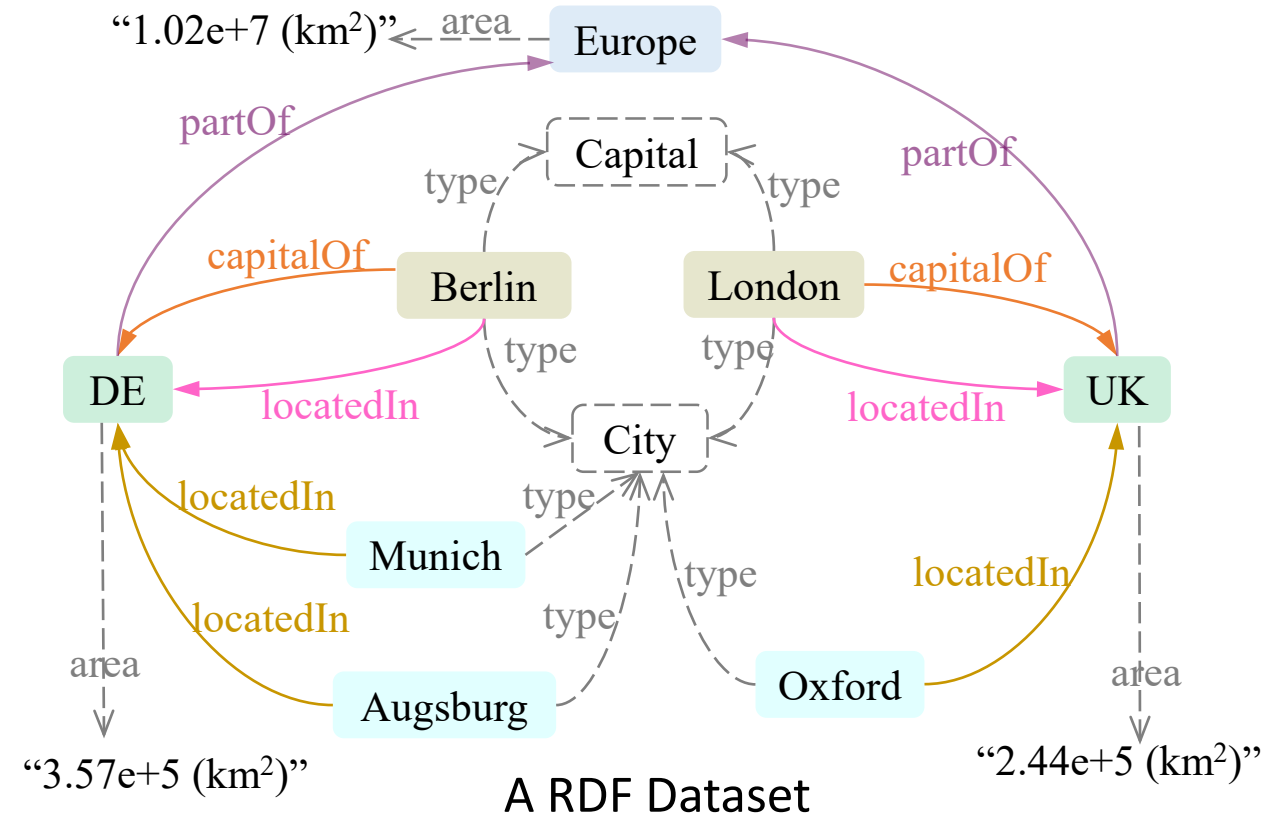


Generation



# Problem Formulation

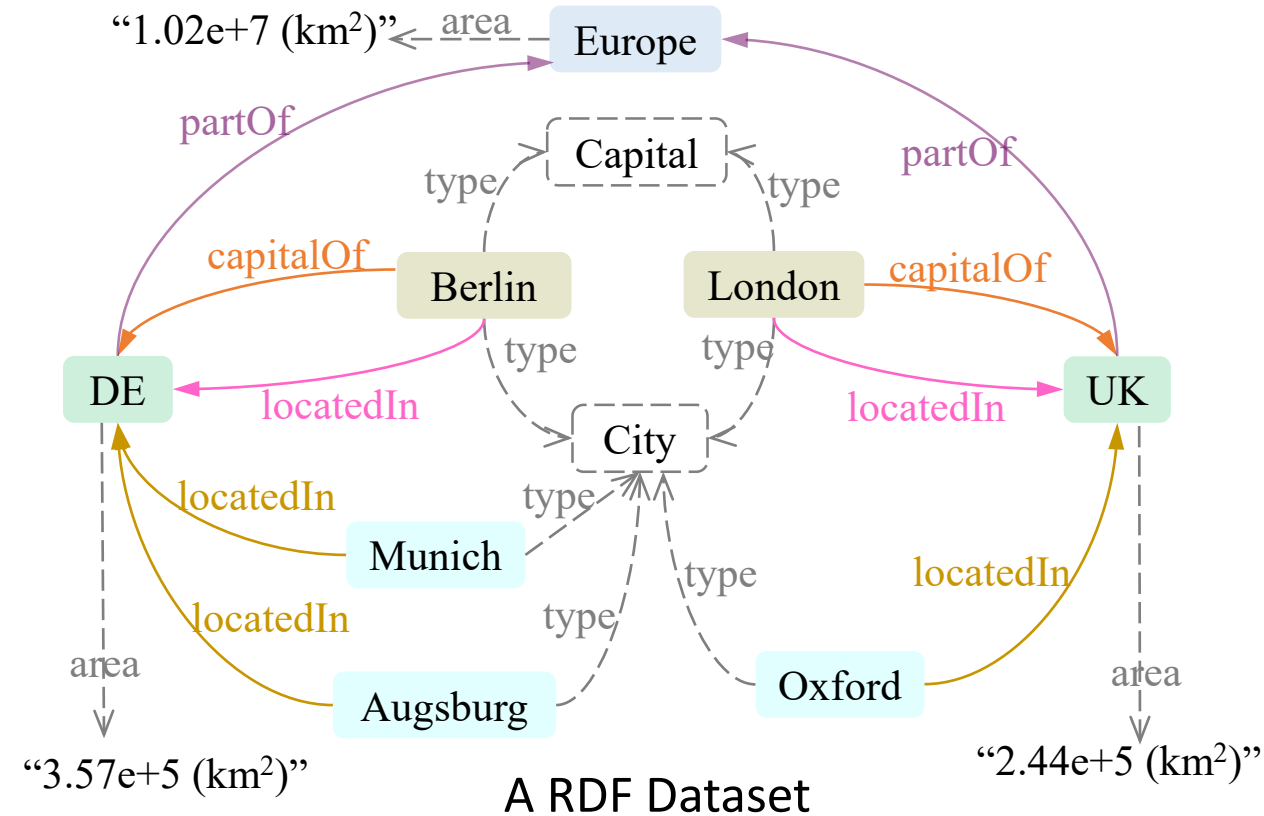
Given a RDF dataset:



# Problem Formulation

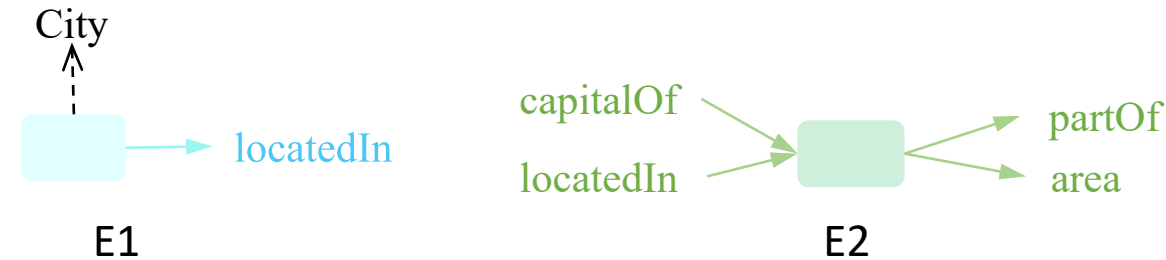
Given a RDF dataset:

- **Entity Description Pattern (EDP)**: the set of classes, forward properties and backward properties



A RDF Dataset

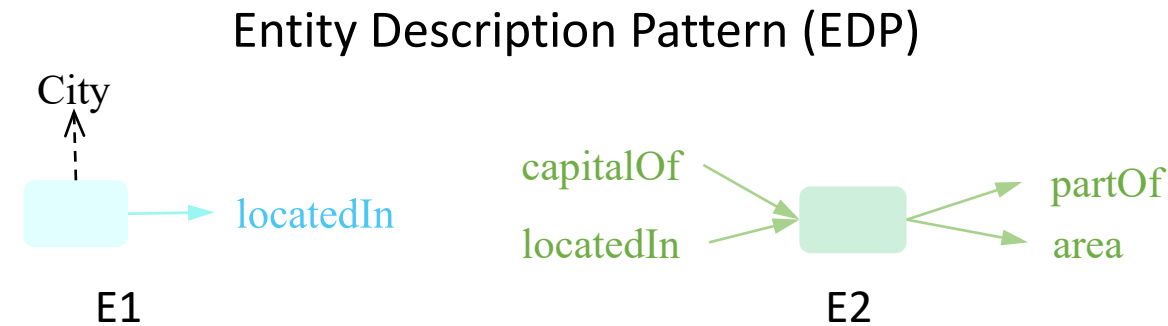
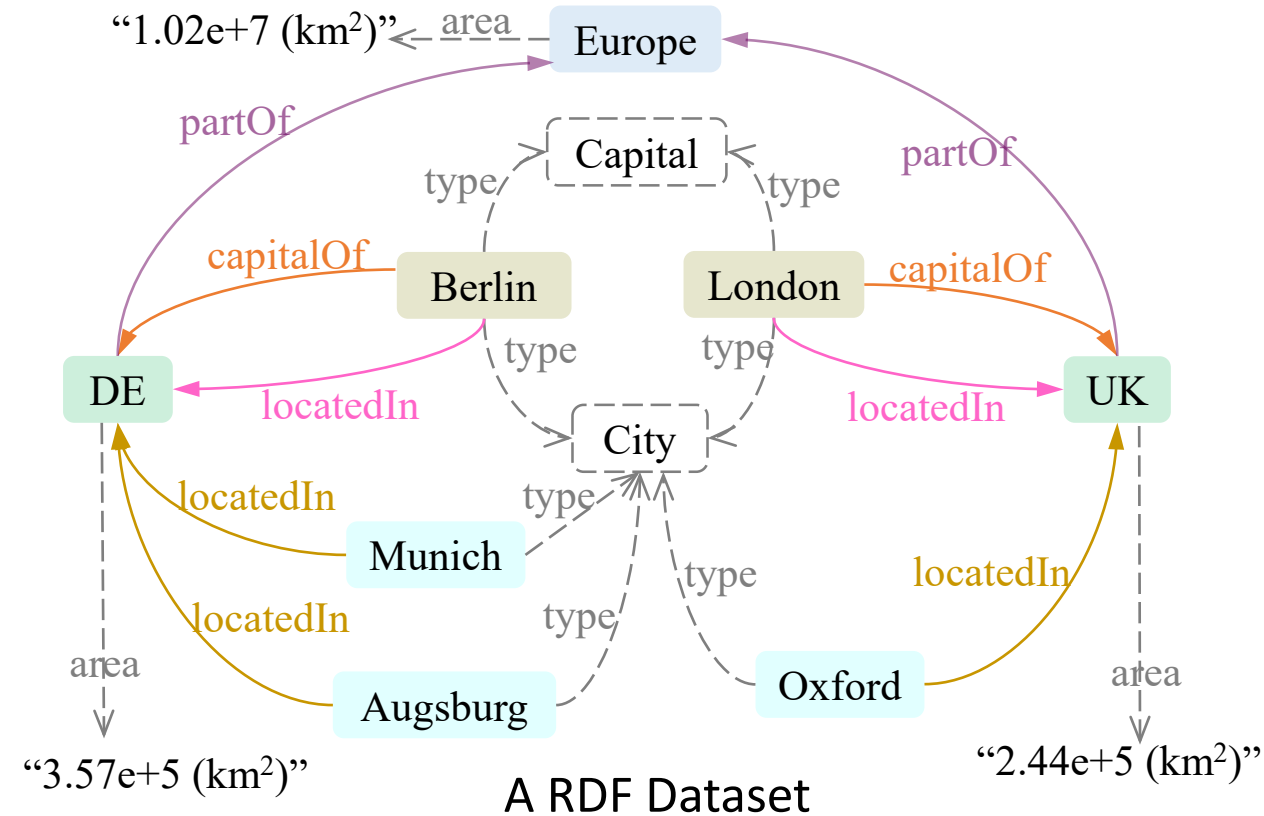
## Entity Description Pattern (EDP)



# Problem Formulation

Given a RDF dataset:

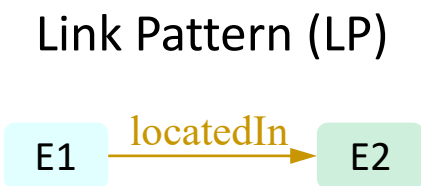
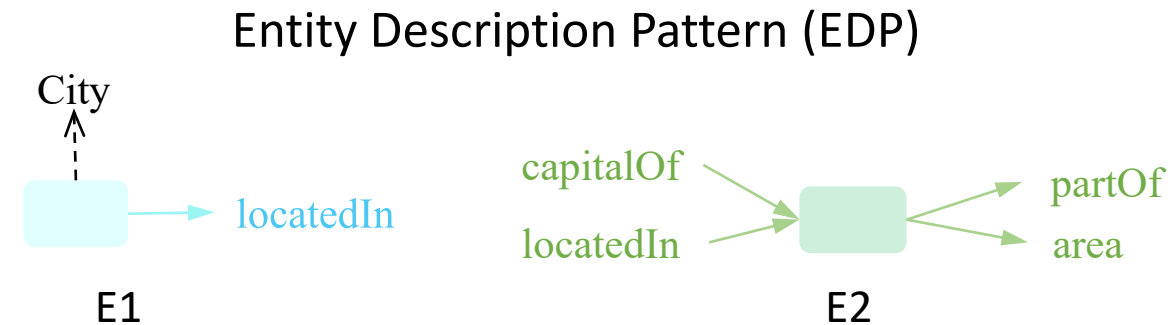
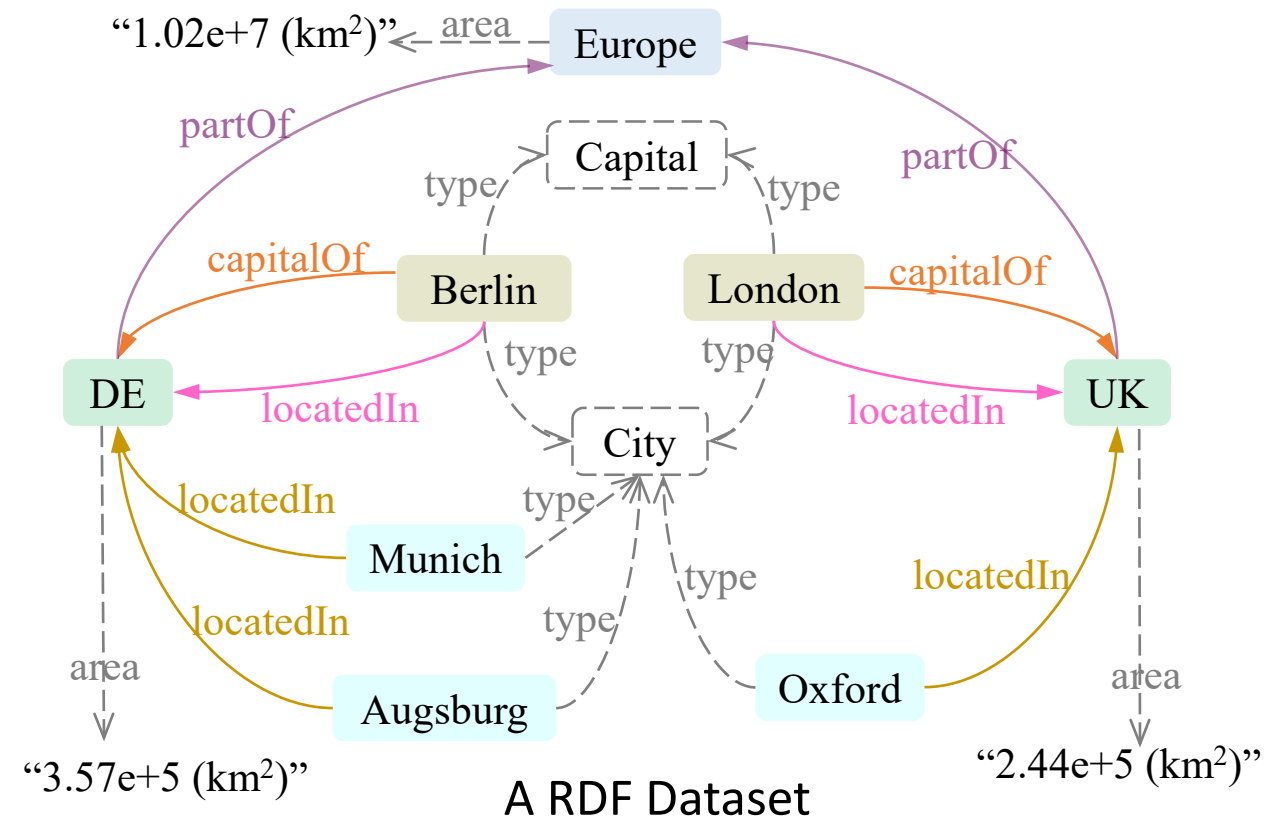
- **Entity Description Pattern (EDP)**: the set of classes, forward properties and backward properties
- **Link Pattern (LP)**: the property and the two EDPs linked by it



# Problem Formulation

Given a RDF dataset:

- **Entity Description Pattern (EDP)**: the set of classes, forward properties and backward properties
- **Link Pattern (LP)**: the property and the two EDPs linked by it
- **Pattern-coverage snippet**: a subgraph covering all EDPs and LPs



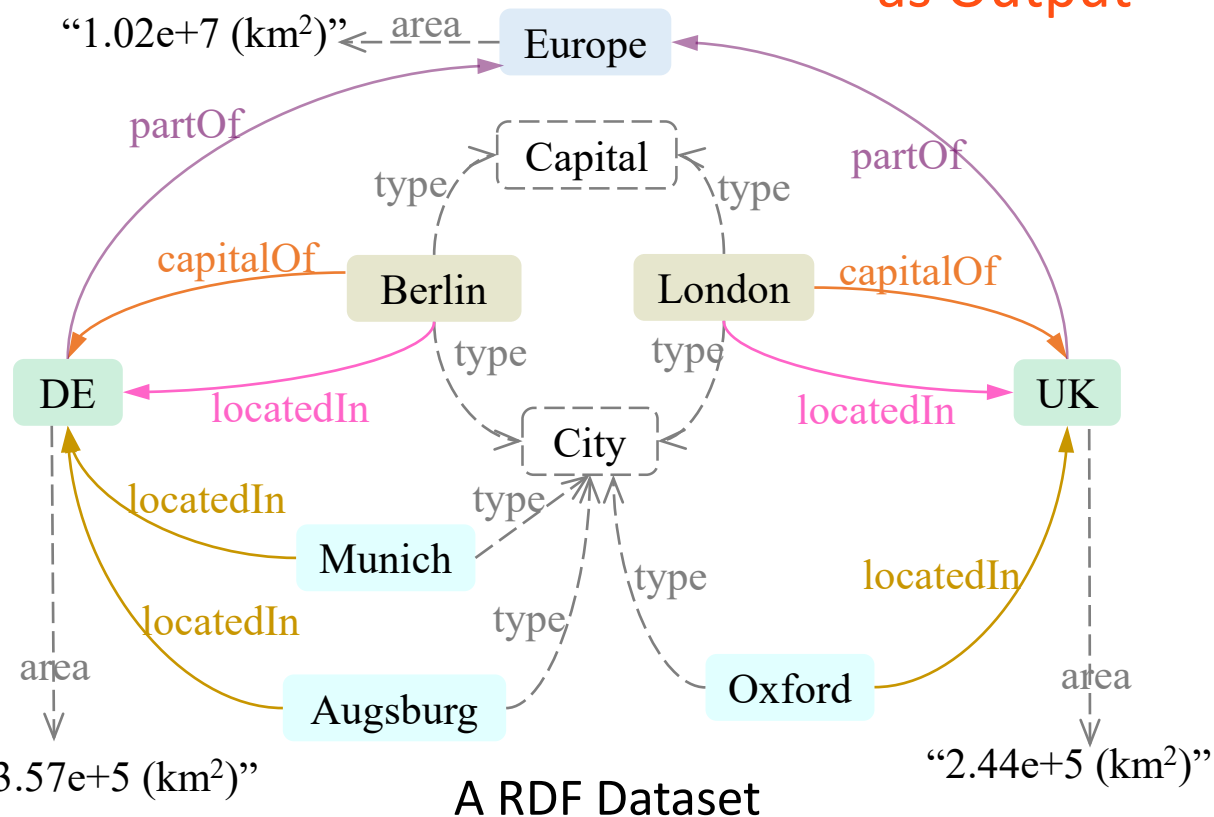


# Problem Formulation

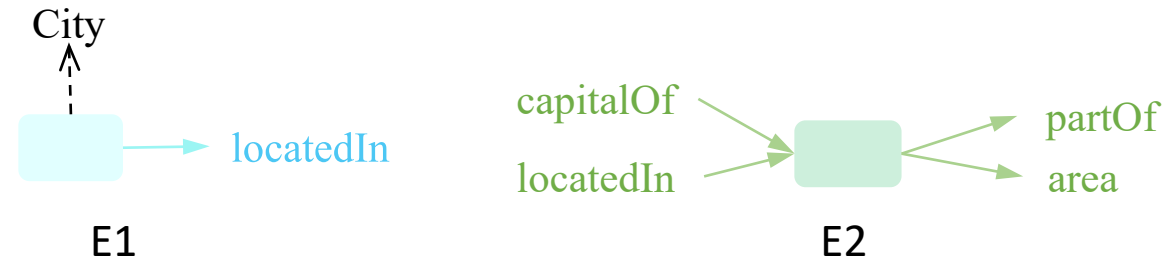
Given a **RDF dataset**:    as Input

- **Entity Description Pattern (EDP)**: the set of classes, forward properties and backward properties
- **Link Pattern (LP)**: the property and the two EDPs linked by it
- **Pattern-coverage snippet**: a subgraph covering all EDPs and LPs

   as Output



Entity Description Pattern (EDP)



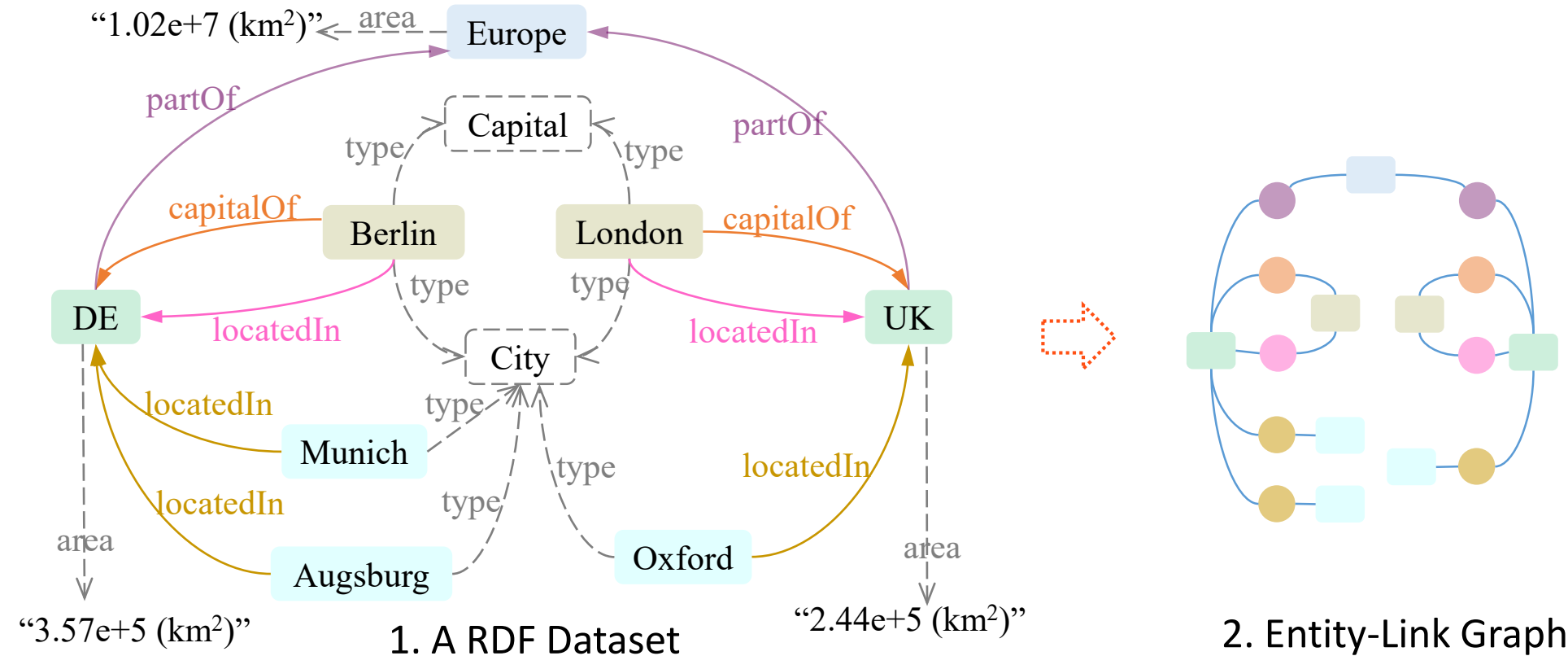
Link Pattern (LP)



# Methods - Alg. Basic

For a connected RDF graph:

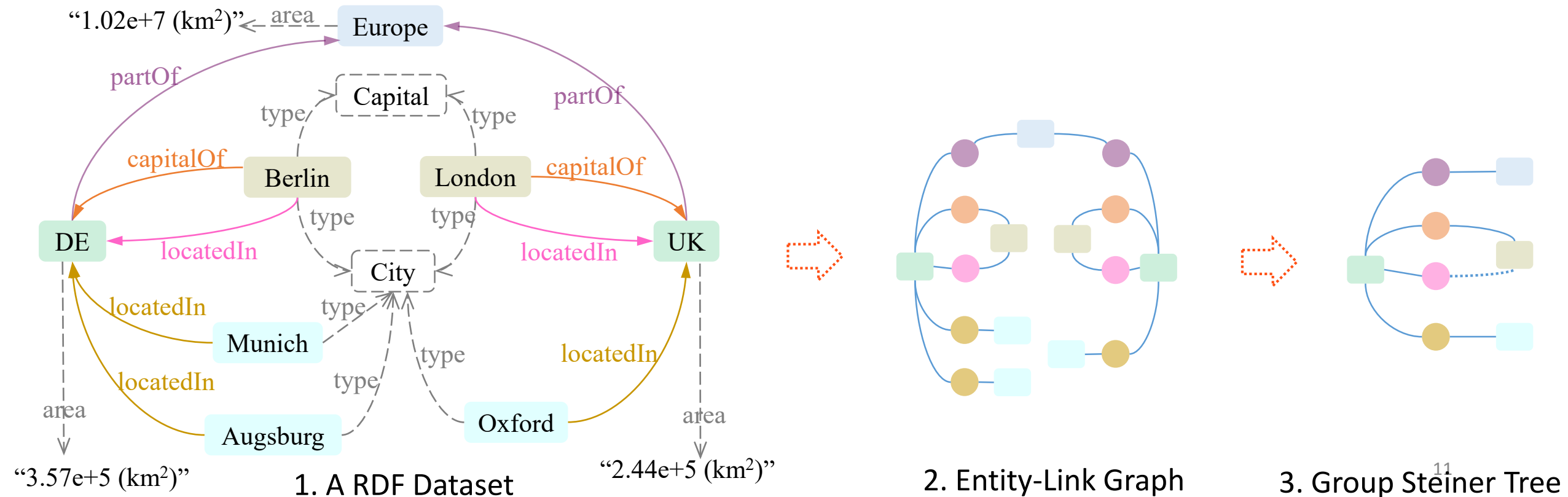
- Each entity is mapped to an EDP, each property (link of two entities) is mapped to a LP



# Methods - Alg. Basic

For a connected RDF graph:

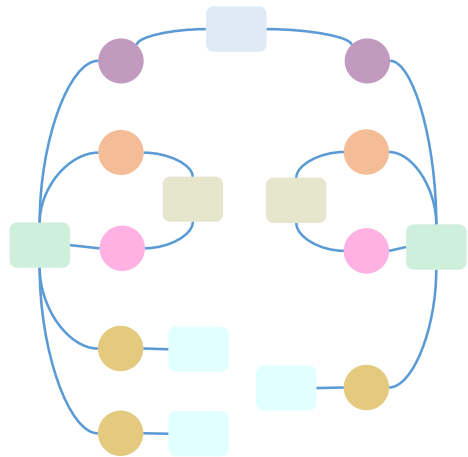
- Each entity is mapped to an EDP, each property (link of two entities) is mapped to a LP
  - Group Steiner Tree Problem: given a set of keys, finding a minimum tree containing all these keys
- Using all EDPs and LPs as keys, to find a GST as snippet



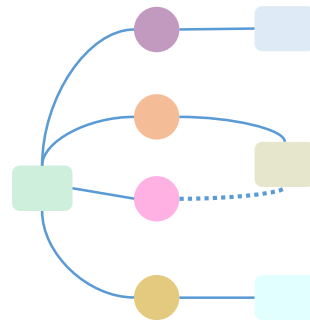
# Methods - Alg. Basic

For a connected RDF graph:

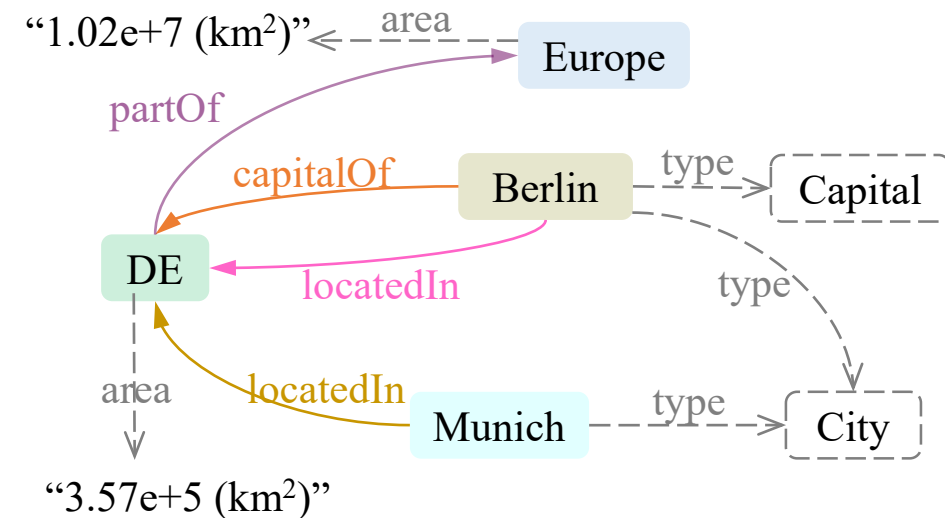
- Each entity is mapped to an EDP, each property (link of two entities) is mapped to a LP
  - Group Steiner Tree Problem: given a set of keys, finding a minimum tree containing all these keys
- Using all EDPs and LPs as keys, to find a GST as snippet



2. Entity-Link Graph



3. Group Steiner Tree



4. Pattern-Coverage Snippet<sup>12</sup>

# Methods - Alg. *PCSG* and Alg. *QPCSG*

---

## Research Questions:

- ✓ 1. How to generate **pattern-coverage** snippets?
2. How to jointly consider **all connected components**?
3. How to be biased towards **keyword queries**?

## Contributions:

- Alg. *Basic*: solving a Group Steiner Tree problem
- Alg. *PCSG*: merging *Basic* results among components
- Alg. *QPCSG*: extending *PCSG* to keywords

# Methods - Alg. PCSG and Alg. QPCSG

## Research Questions:

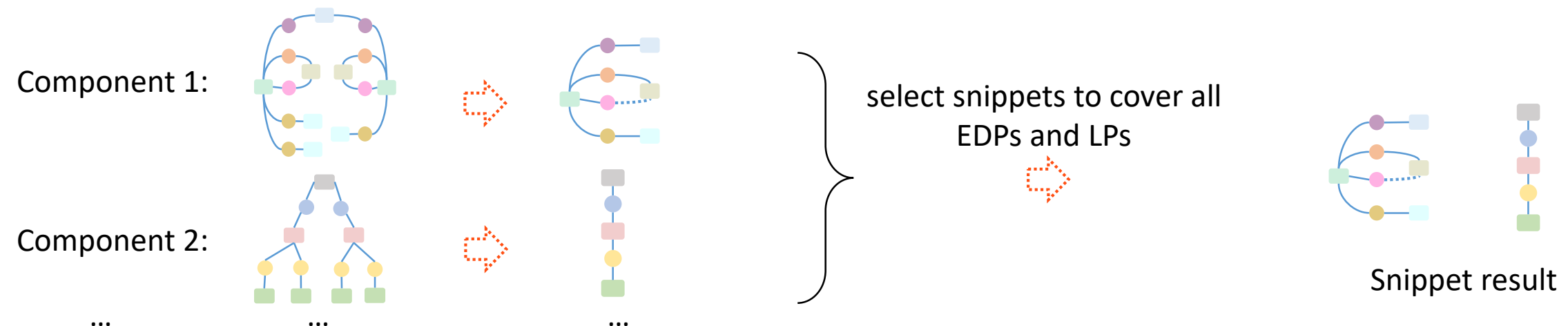
- ✓ 1. How to generate **pattern-coverage** snippets?
- ✓ 2. How to jointly consider **all connected components**?
3. How to be biased towards **keyword queries**?

## Contributions:

- Alg. **Basic**: solving a Group Steiner Tree problem
- Alg. **PCSG**: merging **Basic** results among components
- Alg. **QPCSG**: extending **PCSG** to keywords

## PCSG:

- Executing **Basic** on each connected component, then merge results as a **Set Cover problem**
- Extended to **PCSG- $\tau$**  for **highly heterogeneous graphs** as a relaxation to cover only frequent patterns



# Methods - Alg. PCSG and Alg. QPCSG

## Research Questions:

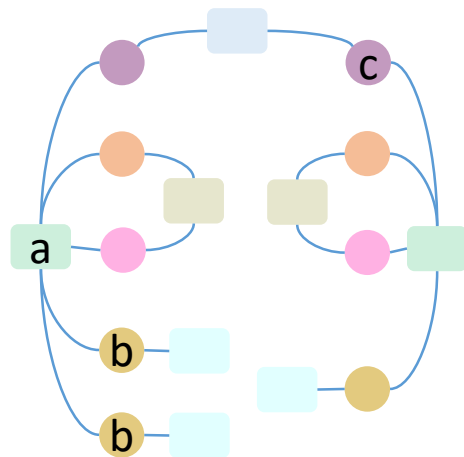
- ✓ 1. How to generate **pattern-coverage** snippets?
- ✓ 2. How to jointly consider **all connected components**?
- ✓ 3. How to be biased towards **keyword queries**?

## Contributions:

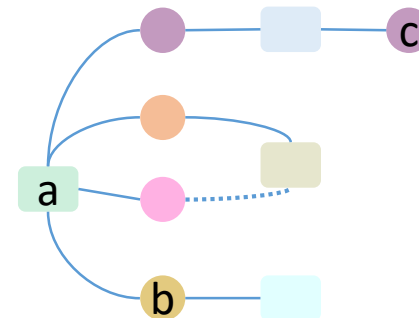
- Alg. **Basic**: solving a Group Steiner Tree problem
- Alg. **PCSG**: merging **Basic** results among components
- Alg. **QPCSG**: extending **PCSG** to keywords

## QPCSG:

- Apart from EDPs and LPs, also **adding query keywords as keys** in the Group Steiner Tree



e.g., Keywords: a, b, c



Snippet result

The result tree should not only cover all EDPs and LPs, but also all the keywords

# Experiments

Datasets: 9,544 real-world RDF datasets from two open data portals

Portal	#RDF datasets	#triples		#classes		#properties		#EDPs		#LPs	
		Median	Max	Median	Max	Median	Max	Median	Max	Median	Max
DataHub.io	311	1,272	20,968,879	3	2,030	16	3,982	15	270,224	27	156,722
Data.gov	9,233	4,000	6,343,524	1	2	13	545	3	500	2	1,103
Overall	9,544	4,000	20,968,879	1	2,030	14	3,982	3	270,224	2	156,722



# Experiments

Datasets: **9,544 real-world RDF datasets** from two open data portals

Portal	#RDF datasets	#triples		#classes		#properties		#EDPs		#LPs	
		Median	Max	Median	Max	Median	Max	Median	Max	Median	Max
DataHub.io	311	1,272	20,968,879	3	2,030	16	3,982	15	270,224	27	156,722
Data.gov	9,233	4,000	6,343,524	1	2	13	545	3	500	2	1,103
Overall	9,544	4,000	20,968,879	1	2,030	14	3,982	3	270,224	2	156,722

Exp-1:

- Compare with SOTA snippet generation methods **by automatic metrics**

Exp-2:

- Compare with SOTA snippet generation methods **by human ratings**

Exp-3:

- Compare with RDF summaries in **assisting human user with SPARQL query completion**

# Experiment - 1

Compare with SOTA snippets **by automatic metrics**:

- Baseline:

IlluSnip extracts a connected snippet containing **frequent classes, properties** and **central entities**

Results:

- $PCSG(-\tau)$  achieved **higher coverage rates**, with **shorter running time**

Table 2: Schema coverage rates (mean  $\pm$  SD).

	Class	Property	EDP	LP
PCSG	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
IlluSnip	0.993 $\pm$ 0.079	0.999 $\pm$ 0.011	0.822 $\pm$ 0.285	0.790 $\pm$ 0.320
PCSG-90%	0.999 $\pm$ 0.019	0.999 $\pm$ 0.006	0.981 $\pm$ 0.030	0.976 $\pm$ 0.035
IlluSnip-90%	0.991 $\pm$ 0.092	0.999 $\pm$ 0.013	0.794 $\pm$ 0.310	0.762 $\pm$ 0.344
PCSG-80%	0.999 $\pm$ 0.025	0.998 $\pm$ 0.010	0.957 $\pm$ 0.061	0.947 $\pm$ 0.071
IlluSnip-80%	0.982 $\pm$ 0.131	0.998 $\pm$ 0.017	0.784 $\pm$ 0.317	0.751 $\pm$ 0.353

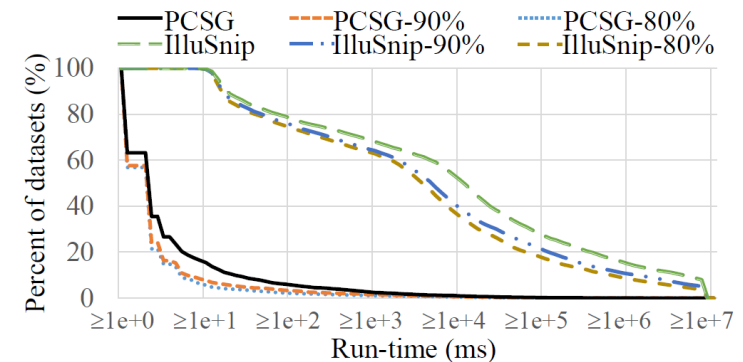


Fig. 6: Cum. distributions of run-time.

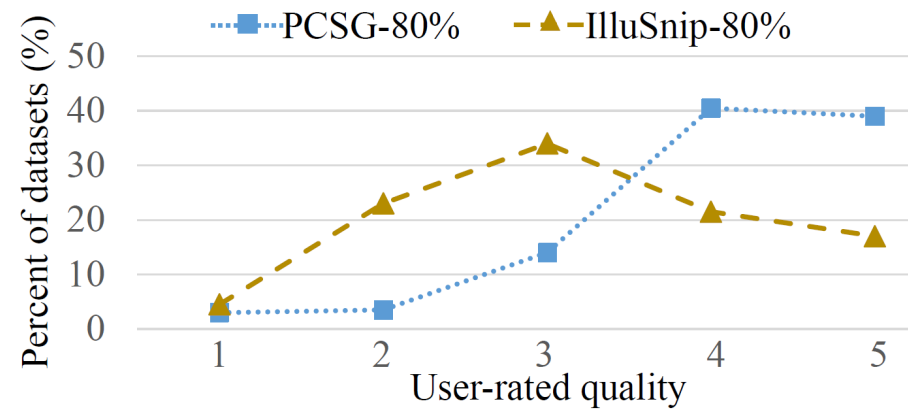
# Experiment - 2

Compare *PCSG* with *IlluSnip* by human ratings:

- Randomly given a dataset with its metadata, classes, properties, EDPs and LPs
- Invite human users to rate two snippets on a 1 – 5 scale

Results:

- On 200 cases, *PCSG* got **higher average score** than *IlluSnip*



# Experiment - 3

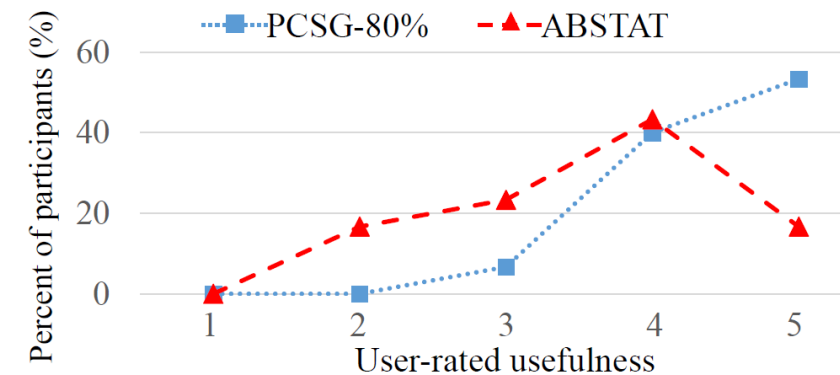
Compare *PCSG* with RDF Summary *ABSTAT*:

- *ABSTAT*: providing triple patterns as summary
- Task: **Completing SPARQL query patterns** assisted by the summary or snippet
- Dataset: DBpedia 2016-10

Results:

- With *PCSG*, users achieved **higher accuracy** in **shorter time**, giving **higher rating of satisfaction**

	Accuracy			Time (seconds)		
	Simple Task	Complex Task	Overall	Simple Task	Complex Task	Overall
PCSG-80%	0.900 ± 0.300	0.900 ± 0.300	0.900 ± 0.300	85.9 ± 39.0	164.0 ± 84.6	124.9 ± 76.6
ABSTAT	0.933 ± 0.249	0.833 ± 0.373	0.883 ± 0.321	117.6 ± 51.1	214.3 ± 154.1	166.0 ± 124.6



# Conclusion

---

1. Presenting snippet generation methods *PCSG* and *QPCSG*
2. Demonstrating their *effectiveness and timeliness*
3. Receiving *higher scores of user rating* and comparison with summary method

In the future:

- Better visualization methods for snippets
- Comprehensive comparison between snippets and summaries

Thanks for your time!

Q & A

Contact: [xxwang@smail.nju.edu.cn](mailto:xxwang@smail.nju.edu.cn)