SPOOL: A SPARQL-based ETL Framework for OLAP over Linked Data

Takahiro Komamizu, Toshiyuki Amagasa, Hiroyuki Kitagawa

University of Tsukuba, Japan

Background: Proliferation of Linked Data

- Linked Data (LD) has been used in various domains.
  - Publishing and connecting data on the Web.
  - Datahub[1] contains more than 10,000 datasets.
  - Linked Open Data Cloud[2] reports more than 1,000 domains.
- Many LD datasets hold useful numerical values.
  - Population
  - Food consumption
  - Money usage
  - etc.

[1] http://datahub.io/
[2]http://lod-cloud.net/

Motivation: Analytical processing

- Analyzing numerical data on LD datasets can reveal important facts.
  - LD datasets have more complicated structures (i.e. graph structure).
  - Various ontologies are used for each dataset.
- Preparation for analysis is laborious.
  - Understanding structures of target datasets.
  - Extracting necessary information for analyses.
  - Developing analytical processors.

Research purpose 1: reduce this effort

Motivation: Large LD datasets

- Previous work [3] tried to achieve the purpose.
  - ETL framework for OLAP over LD datasets.
  - Processing LD datasets in local servers.
  - The algorithm requires to read all data.
- Problems
  - Downloading large datasets takes long time.
  - Reading all data is inefficient.

Research purpose 2: efficient processing

[3] Inoue et al., An ETL Framework for Online Analytical Processing of Linked Open Data, WAIM 2013

Objective

- Achieve the purposes
  - Reducing efforts for preparation of analyses.
  - Efficiently processing large LD datasets.
- Overcome the previous work [3]
  - Aiming at enabling OLAP for LD datasets
    - OLAP is typical and powerful analytical processing paradigm.
  - Processing datasets w/o downloading whole datasets.
  - Formally defining ETL process for LD datasets (cf. paper).

[3] Inoue et al., An ETL Framework for Online Analytical Processing of Linked Open Data, WAIM 2013

SPOOL: proposed framework

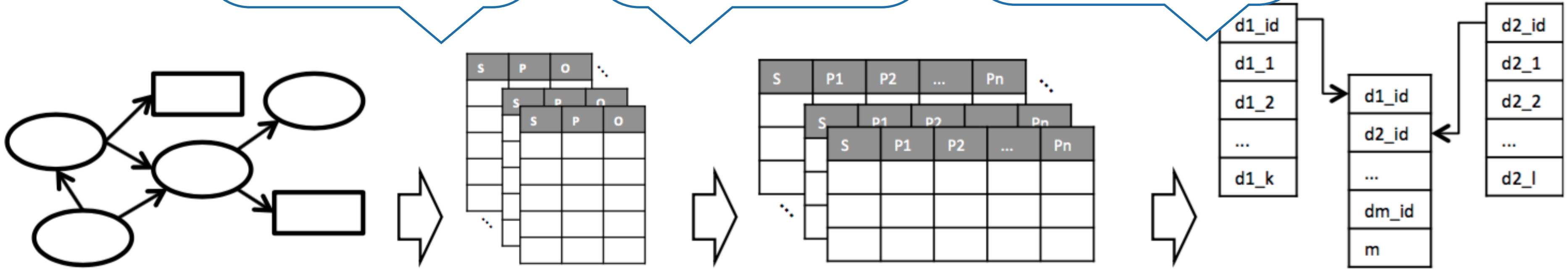
- Idea
    - Extracting only necessary information through SPARQL endpoints of LD datasets.
    - Utilizing search engine optimization on SPARQL endpoints.
- Components
    - SPARQL-based Type-partitioned Triple Store (TPTS)
        - TPTS: extracting OLAP-related information for LD datasets
            - This process is originally offline process.
        - A series of SPARQL queries to construct TPTS.

TPTS approach[3]: overview

Extracting attributes and values for each class

Reconstructing TPTSs into record-based tables (PT)

Generating star/snowflake schemas for OLAP



(a) Input LD dataset.

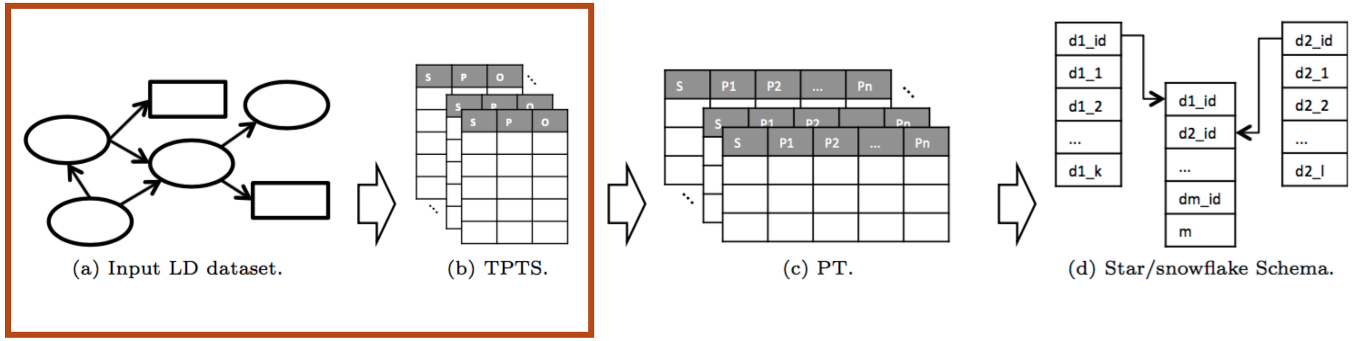
(b) TPTS.

(c) PT.

(d) Star/snowflake Schema.

[3] Inoue et al., An ETL Framework for Online Analytical Processing of Linked Open Data, WAIM 2013

TPTS approach: TPTS extraction

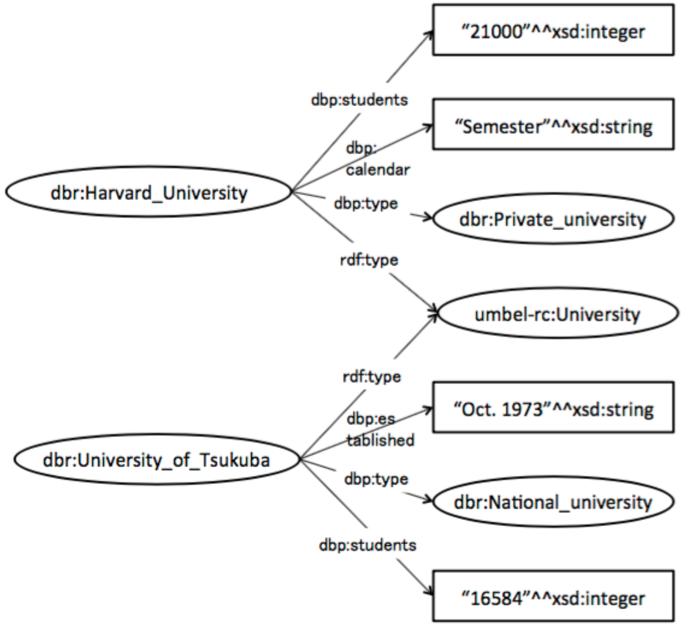


(a) Input LD dataset.

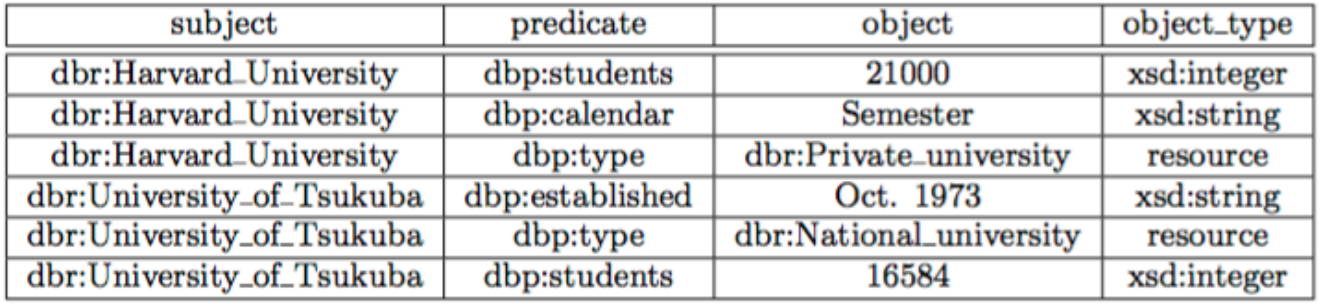
(b) TPTS.

(c) PT.

(d) Star/snowflake Schema.

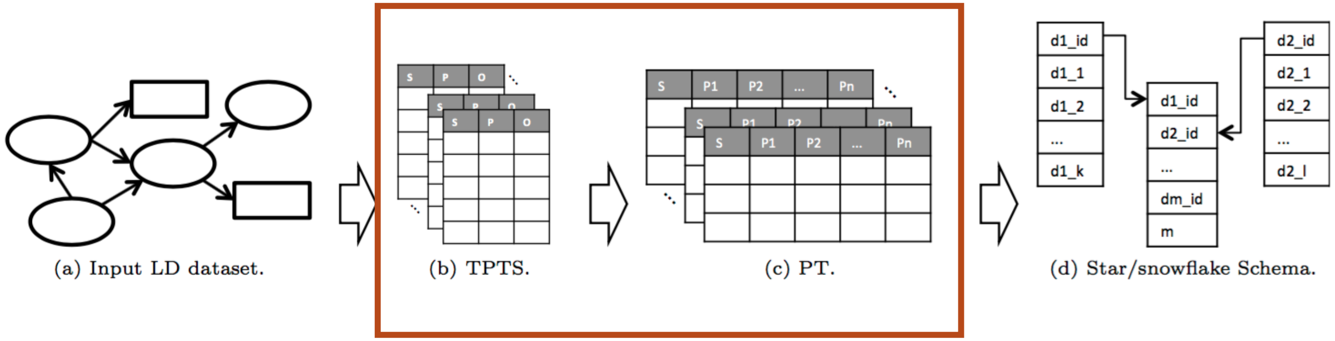


- Extract triples which subject is of a class (identifying rdf:type)
  - e.g., umbel-rc:University



subject	predicate	object	object_type
dbr:Harvard_University	dbp:students	21000	xsd:integer
dbr:Harvard_University	dbp:calendar	Semester	xsd:string
dbr:Harvard_University	dbp:type	dbr:Private_university	resource
dbr:University_of_Tsukuba	dbp:established	Oct. 1973	xsd:string
dbr:University_of_Tsukuba	dbp:type	dbr:National_university	resource
dbr:University_of_Tsukuba	dbp:students	16584	xsd:integer

TPTS approach: PT extraction

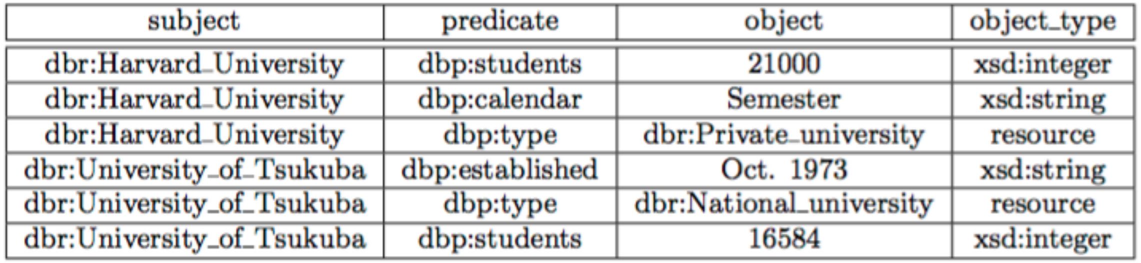


(a) Input LD dataset.

(b) TPTS.

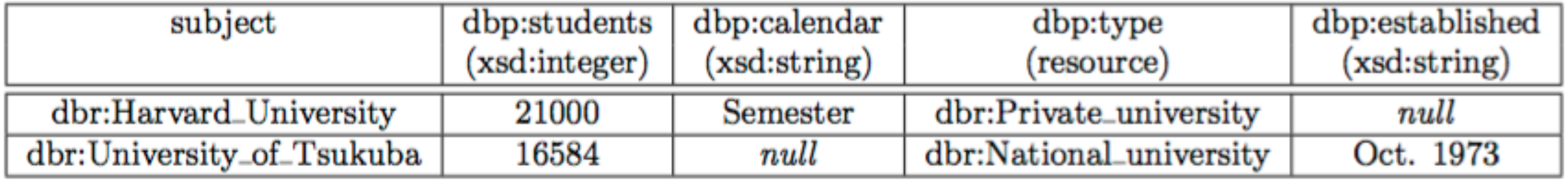
(c) PT.

(d) Star/snowflake Schema.



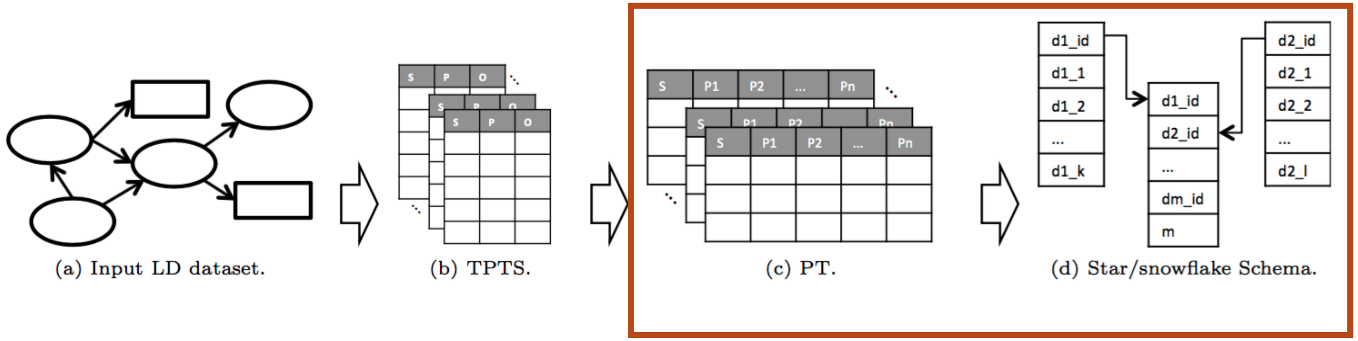
subject	predicate	object	object_type
dbr:Harvard_University	dbp:students	21000	xsd:integer
dbr:Harvard_University	dbp:calendar	Semester	xsd:string
dbr:Harvard_University	dbp:type	dbr:Private_university	resource
dbr:University_of_Tsukuba	dbp:established	Oct. 1973	xsd:string
dbr:University_of_Tsukuba	dbp:type	dbr:National_university	resource
dbr:University_of_Tsukuba	dbp:students	16584	xsd:integer

- Reconstruct tables in TPTS into record-level tables (or PTs) using distinct predicates



subject	dbp:students (xsd:integer)	dbp:calendar (xsd:string)	dbp:type (resource)	dbp:established (xsd:string)
dbr:Harvard_University	21000	Semester	dbr:Private_university	*null*
dbr:University_of_Tsukuba	16584	*null*	dbr:National_university	Oct. 1973

TPTS approach: Schema generation



(a) Input LD dataset.

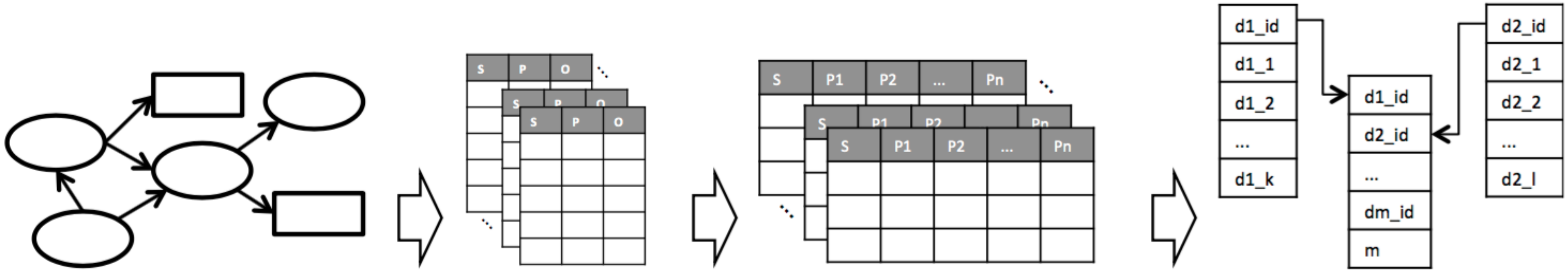
(b) TPTS.

(c) PT.

(d) Star/snowflake Schema.

- Generating star/snowflake schema from PTs.
  - From PT, attributes for each class are obtained.
  - An attribute is specified as measure for OLAP.
  - Other attributes in the same table as measure are considered as dimensions.

SPOOL framework: idea

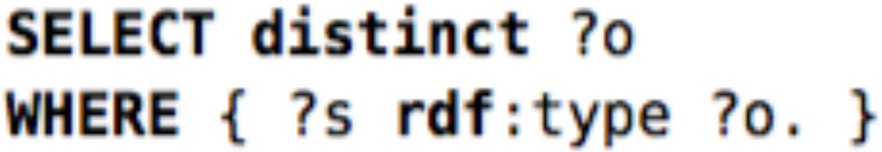


(a) Input LD dataset. (b) TPTS. (c) PT. (d) Star/snowflake Schema.

- Classes and attributes are required for determining PT structures.
- Instances of classes are required during PT construction.

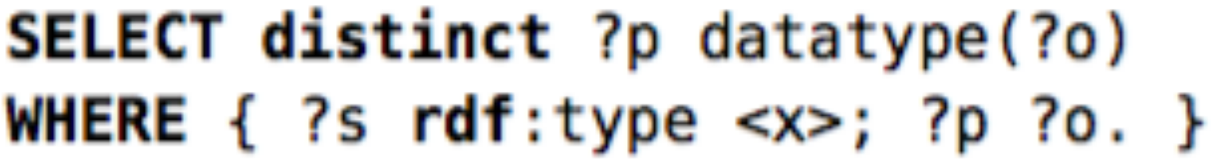
SPOOL framework: class and attributes

- Obtaining classes

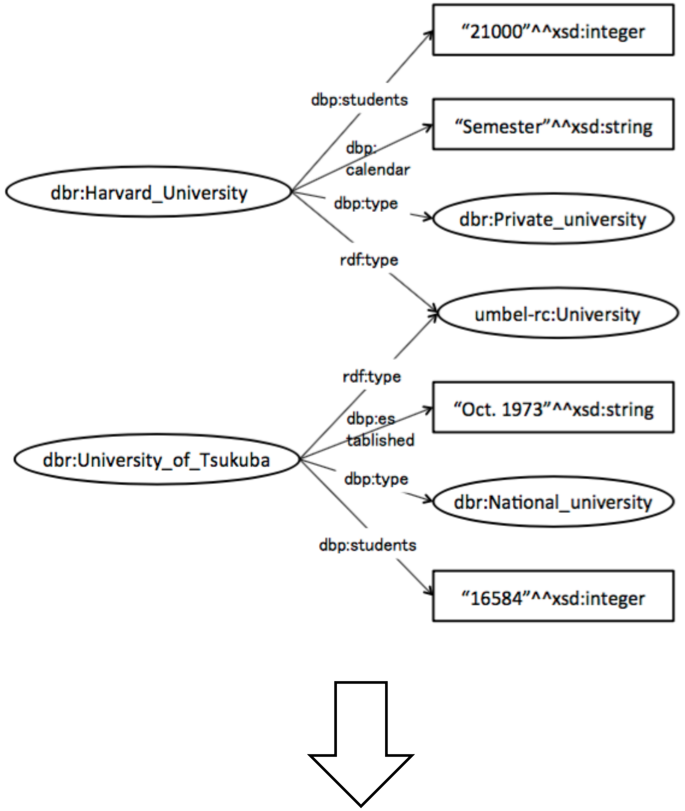


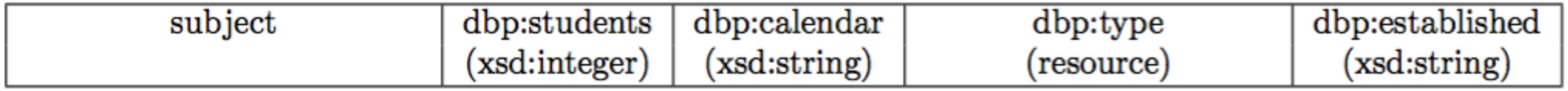
```
SELECT distinct ?o
WHERE { ?s rdf:type ?o. }
```

- Obtaining attributes of a class x



```
SELECT distinct ?p datatype(?o)
WHERE { ?s rdf:type <x>; ?p ?o. }
```

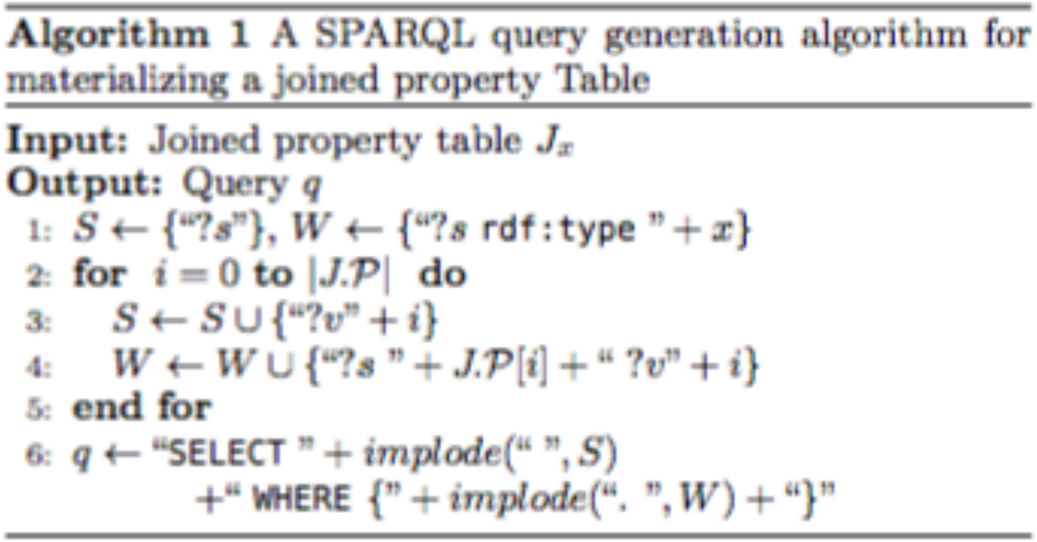




subject	dbp:students (xsd:integer)	dbp:calendar (xsd:string)	dbp:type (resource)	dbp:established (xsd:string)

SPOOL framework: materialization

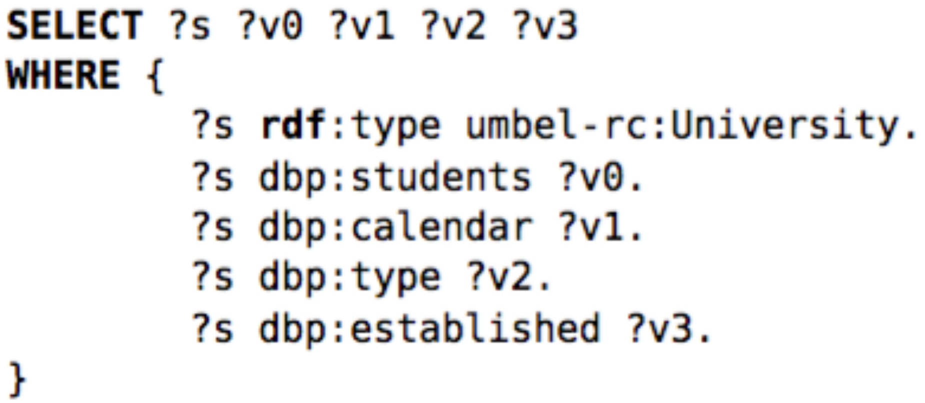
- In this step, values of attributes for instances of classes are extracted.



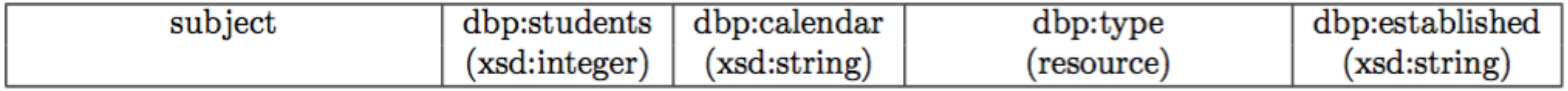
Algorithm 1 A SPARQL query generation algorithm for materializing a joined property Table

Input: Joined property table J_x
Output: Query q

```
$S \leftarrow \{\text{"?s"}\}$, $W \leftarrow \{\text{"?s rdf:type "} + x\}$
for $i = 0$ to $|J.\mathcal{P}|$ do
    $S \leftarrow S \cup \{\text{"?v"} + i\}$
    $W \leftarrow W \cup \{\text{"?s "} + J.\mathcal{P}[i] + \text{" ?v"} + i\}$
end for
$q \leftarrow \text{"SELECT "} + implode(\text{" "}, S)$
        $+ \text{" WHERE \{"} + implode(\text{". "}, W) + \text{"\}"}$
```



```
SELECT ?s ?v0 ?v1 ?v2 ?v3
WHERE {
        ?s rdf:type umbel-rc:University.
        ?s dbp:students ?v0.
        ?s dbp:calendar ?v1.
        ?s dbp:type ?v2.
        ?s dbp:established ?v3.
}
```



subject	dbp:students (xsd:integer)	dbp:calendar (xsd:string)	dbp:type (resource)	dbp:established (xsd:string)

Empirical study

- Purpose: Check applicability of SPOOL framework.
- Datasets
  - CIA World Factbook [4]
  - DBpedia [5]
- Methodology
  - Apply SPOOL to these datasets and observe the outputs.
- Results (cf. paper) indicate
  - Applicability of SPOOL framework for LD datasets.

[4] http://wifo5- 03.informatik.uni- mannheim.de/factbook/snorql/
[5] http://dbpedia.org/sparql

Related work: OLAP for LD

- Dedicated ontology based approaches (e.g., [Kaempgen et al. 2011])
    - Dedicated vocabularies indicate which parts of LD datasets form OLAP cubes.
    - Vocabularies are RDF Data Cube vocabulary, Open Cube vocabulary, and their extensions.
- Human-supported ETL (e.g., [Niinimaki et al. 2009])
    - This kind of approach determines mapping from an LD dataset to OLAP schema with help of users.

Conclusion

- SPOOL framework
    - OLAP schema and instances extraction from SPARQL endpoints of LD datasets.
    - Advantages
        - No need to download whole datasets in advance.
        - Utilizing search engine performance on SPARQL endpoints.
        - Small amount of human efforts is required.
- Future work
    - Enrichment of dimension hierarchy using external vocabularies.
    - Update mechanism.
    - Missing **rdf:type** situations.