Schema Linking

Juan Carlos Rosito Cuellar

Universita' degli studi di Milano-Bicocca

16-4-2019

The scope of the challenge

Semantic Web Challenge on Tabular Data to Knowledge Graph Matching

- Tabular data in the form of CSV files is the common input format in a data analytics pipeline. However a lack of understanding of the semantic structure and meaning of the content may hinder the data analytics process. Thus gaining this semantic understanding will be very valuable for data integration, data cleaning, data mining, machine learning and knowledge discovery tasks.
- This challenge aims at benchmarking systems dealing with the tabular data to KG matching problem, so as to facilitate their comparison on the same basis and the reproducibility of the results.

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

objectives

- Recommend a schema semantical annotation of a table
  - Get Type Candidates based on the instances of the table
  - Get Predicate Candidates based on the Types of each Column of the table
  - Generate a Score of all Types and Columns
  - Generate all possible Ontologies with the candidates of Type and Predicate.
  - Generate a Score of all possible Ontologies.
  - Choose the best Schema semantical annotation of the Table

Introduction: Definitions

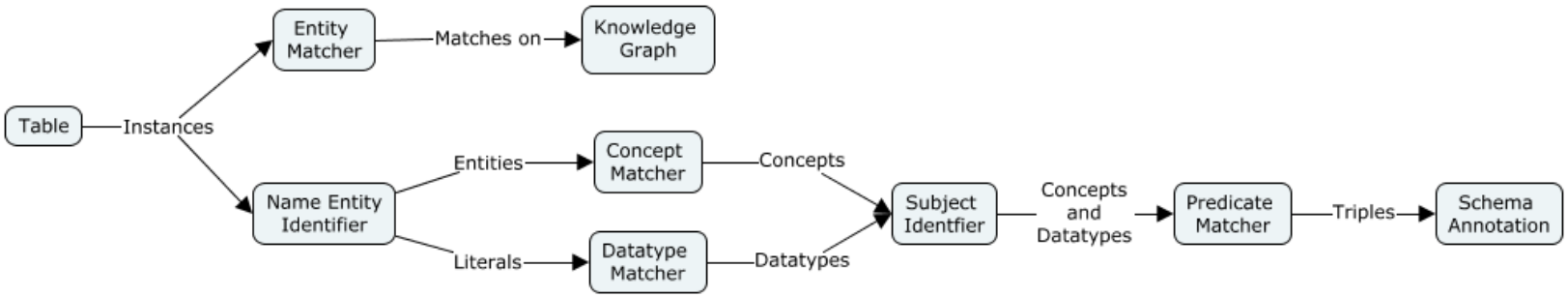


Figure 1: Tabular overview of the variables

The inputs of table interpretation: TABLE

Definition (Table)

We define *Table* a rectangular array (matrix) of strings arranged in n rows and m columns. Every pair (i, j) with $1 \leq i \leq n$ and $1 \leq j \leq m$, is unambiguously identifies a *cell* of the table.

Definition (Rows and Columns)

Given an $n \times m$ table, let r_i denote the i-th row of the table, that is $r_i = \{(i, j) | 1 \leq j \leq m\}$ and c_j denote the j-th column $(c_j = \{(i, j) | 1 \leq i \leq n\})$. Let $\mathcal{R} = \{r_i | 1 \leq j \leq m\}$ and $\mathcal{C} = \{c_j | 1 \leq i \leq n\}$ be the set of all rows and columns of the table, respectively.

The inputs of table interpretation: TABLE2

Definition (Column header function)

A Column Header Function $h : \mathcal{C} \rightarrow \mathcal{L}$ associates each column with a word of a language $\mathcal{L}$.

Definition (Header Table)

We define a Header Table the pair $T_h = (T, h)$ where T is a table and h is a column header function.

Definition (Header)

Given a header table $T_h = (T, h)$, we define $\mathcal{H} = h(\mathcal{C})$ as the header of table T.

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

The inputs of table interpretation: Semantic

Definition (Ontology - simplified definition)

An ontology is a multigraph $\mathcal{O} = (\mathcal{N}, \mathcal{P}, \mathcal{A})$, where:

- $\mathcal{N} = \mathcal{N}_c \cup \mathcal{N}_d$ is the set of the entities in $\mathcal{O}$ (e.g., DBpedia Ontology, GeoNames Ontology, ...)
  - $\mathcal{N}_c$ is the set of concepts (e.g., dbo:Movie, dbo:Actor, ...)
  - $\mathcal{N}_d$ is the set of data types (e.g., xsd:date, xsd:integer, ...)
- $\mathcal{P}$ is the property label set (e.g., "starring", "releaseDate", ...)
- $\mathcal{A} \subset \mathcal{N}^2 \times \mathcal{P}$ set of labeled directed arcs, where an edge can exist only between concepts or between a concept and a data type.

The inputs of table interpretation: Semantic 2

Definition (Knowledge Graph)

Given an ontology $\mathcal{O} = (\mathcal{N}, \mathcal{P}, \mathcal{A})$, a *Knowledge Graph* $\mathcal{KG}$ [**?**] is a directed multigraph defined by the tuple $\mathcal{KG} = (\mathcal{V}, \mathcal{E}, \mathcal{O}, \psi, \phi)$ where:

- $\mathcal{V}$ is the set of vertices; a vertex represents an entity or a literal (e.g., dbr:The Matrix, dbr:Keanu Reeves, "1999", ...)
- $\mathcal{E} \subset \mathcal{V}^2$ is a set of directed edges connecting two nodes, they represent links between two entities;
- ψ is the ontology mapping function $\psi : \mathcal{V} \to \mathcal{N}_o$, which links an entity vertex to a concept or data type in the ontology (e.g., dbo:Movie links dbr:The Matrix, dbo:Actor links dbr:Keanu Reeves, xsd:date links "1999")
- ϕ is the predicate mapping function $\phi : \mathcal{E} \to \mathcal{E}$, which maps an edge to a predicate type (e.g. dbr:The Matrix dbo:starring dbr:Keanu Reeves, dbr:The Matrix dbo:releaseDate "1999").

The inputs of table interpretation: Formalization

Given an $m \times n$ header table $\mathcal{T} = (T, h)$:

- $T = \{t_{ij} : 1 \leq i \leq n \wedge 1 \leq j \leq m\}$ where t_{ij} is the element contained in the cell (i, j)
- $\mathcal{C} = \{c_1, ..., c_m\}$ where c_j is the j-th column
- $\mathcal{R} = \{r_1, ..., r_n\}$ where r_i is the i-th row

And a *set of Knowledge Graphs* $\mathcal{KG}$s is defined as follows:
$\mathcal{KG}s = \{\mathcal{KG}_1, \mathcal{KG}_2, \ldots, \mathcal{KG}_k\}$

The inputs of table interpretation: Formalization2

And a *Knowledge Graph* $\mathcal{KG}$, defined as follows:
$\mathcal{KG}_x = (\mathcal{V}_x, \mathcal{E}_x, \mathcal{O}_x, \psi_x, \phi_x)$

- $\mathcal{V}_x = \mathcal{Z}_x \cup \mathcal{L}_x$
  - $\mathcal{Z}_x$ is a set of entities in the $\mathcal{KG}_x$
  - $\mathcal{L}_x$ is a set of literals
- $\mathcal{E}_x$ is a set of labeled directed edges between two elements in $\mathcal{N}_x$
- $\mathcal{O}_x = (\mathcal{N}_x, \mathcal{P}_x, \mathcal{A}_x)$ with $\mathcal{N}_x = \mathcal{N}_{cx} \cup \mathcal{N}_{dx}$
  - $\mathcal{N}_{cx}$ is a set of concepts in the $\mathcal{KG}_x$
  - $\mathcal{N}_{dx}$ is a set of datatypes in the $\mathcal{KG}_x$
- $\psi_x : \mathcal{V}_x \rightarrow \mathcal{N}_x$ is the ontology mapping function
- $\phi_x : \mathcal{E}_x \rightarrow \mathcal{A}_x$ is the predicate mapping function

The inputs of table interpretation: Concept

Definition (Concept Matcher)

Given a knowledge base $\mathcal{KG}_x$, the *Concept Matcher* is a function $\theta_x : T \rightarrow \mathcal{N}_{cx} \cup \emptyset$:

$$\eta_x(t_{ij}) = \begin{cases} c \in \mathcal{N}_{cx} \\ \emptyset \end{cases} \quad \forall t_{ij} \in T \tag{1}$$

The inputs of table interpretation: Type

Definition (Datatype Matcher)

Given a knowledge base $\mathcal{KG}_x$, the *Datatype Matcher* is a function $\theta_x : T \rightarrow \mathcal{N}_{dx} \cup \emptyset$:

$$\theta_x(t_{ij}) = \begin{cases} d \in \mathcal{N}_{dx} \\ \emptyset \end{cases} \quad \forall t_{ij} \in T \tag{2}$$

The inputs of table interpretation: Type

Definition (Entity Matcher)

Given a knowledge base $\mathcal{KG}_x$, an *Entity Matcher* is a function $\chi_x : T \rightarrow \mathcal{Z}_x \cup \emptyset$:

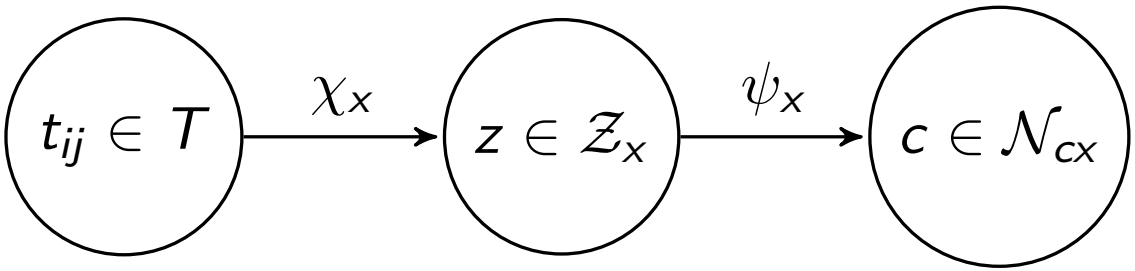
$$\chi_x(t_{ij}) = \begin{cases} z \in \mathcal{Z}_x \\ \emptyset \end{cases} \quad \forall t_{ij} \in T \tag{3}$$

The inputs of table interpretation: Type2

Lemma

Given a knowledge base $\mathcal{KG}_x$ and an Entity Matcher is a function χ_x, a particular Concept Matcher is defined as:

$$\eta_x(t_{ij}) = \begin{cases} \psi_x(\chi_x(t_{i,j})), & \text{if } \chi_x(t_{i,j}) \in \mathcal{Z}_x \\ \emptyset, & \text{otherwise} \end{cases} \quad \forall t_{ij} \in T \qquad (4)$$



The inputs of table interpretation: Entity Identifier

Definition (Name Entity Identifier)

Given a knowledge base $\mathcal{KG}_x$, a *Name Entity Identifier* is function $\alpha_x : \mathcal{C} \rightarrow \{\text{“Name Entity”}, \text{“Literal”}\}$.

Lemma

Given a knowledge base $\mathcal{KG}_x$, a threshold value $\bar{\gamma} \in \mathbb{R}$, and a function $\beta_x : T \rightarrow \{1, 0\}$ defined as follows:

$$\beta_x(t_{ij}) = \begin{cases} 1, & \text{if } \eta_x(t_{i,j}) \in \mathcal{N}_{cx} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

a particular Name Entity Identifier can be defined as:

$$\alpha_x(c_j) = \begin{cases} \text{“Name Entity”}, & \text{if } \sum_i \beta_x(t_{ij}) \geq \bar{\gamma}; \\ \text{“Literal”}, & \text{otherwise.} \end{cases} \quad \forall c_j \in \mathcal{C} \tag{6}$$

The inputs of table interpretation: Subject Identifier

Definition (Subject Identifier)

Given a knowledge base $\mathcal{KG}_x$, an $m \times n$ header table $\mathcal{T} = (T, h)$: we define *Subject Identifier* a function $\sigma_x : \mathcal{T} \rightarrow \{1, 0\}^m$ so that $\sum_{j=1}^{m} \sigma_x(\mathcal{T})_j = 1$ and $\sigma_x(\mathcal{T})_j = 0,\ \forall j \in \{j | \alpha_x(c_j) = 1\}$.

The inputs of table interpretation: Semantic Column Annotator

Definition (Semantic Column Annotator)

Given a knowledge base $\mathcal{KG}_x$, a table defined by a set of columns $\mathcal{C}$ a Semantic Column Annotator is a function $\zeta_x : \mathcal{C} \rightarrow \mathcal{N}_x$.

The inputs of table interpretation: Semantic Column Annotator 2

Lemma

Given a knowledge base $\mathcal{KG}_x$, a table T, a concept matcher η_x, a datatype matcher θ_x, and sets $\mathcal{D}^j_{cx}$ and $\mathcal{D}^j_{dx}$ defined as follows:

$$\mathcal{D}^j_{cx} = \{t_{ij} | t_{ij} \in T \wedge \eta_x(t_{ij}) = cx \wedge i \in \{i, \ldots, n\}\}, \quad \forall cx \in \mathcal{N}_{cx} \wedge \forall j \in \{1$$

$$\mathcal{D}^j_{dx} = \{t_{ij} | t_{ij} \in T \wedge \theta_x(t_{ij}) = dx \wedge i \in \{i, \ldots, n\}\}, \quad \forall dx \in \mathcal{N}_{dx} \wedge \forall j \in \{1$$

The function:

$$\zeta(c_j) = \begin{cases} \arg\max_{c_x \in \mathcal{N}_{cx}} |D^j_{cx,}|, & \text{if } \alpha(c_j) = 1; \\ \arg\max_{d_x \in \mathcal{N}_{dx}} |\mathcal{D}^j_{dx}|, & \text{otherwise.} \end{cases} \tag{7}$$

is a Semantic Column Annotator.

The inputs of table interpretation: Predicate Matcher

Definition (Predicate Matcher)

$\pi_x : \mathcal{C}^2 \rightarrow \mathcal{A}_x \cup \emptyset$ with $\pi_x(c_i, c_j) = \emptyset$, $\forall (i,j) | i = j$.

Methodology: Pipe-Line

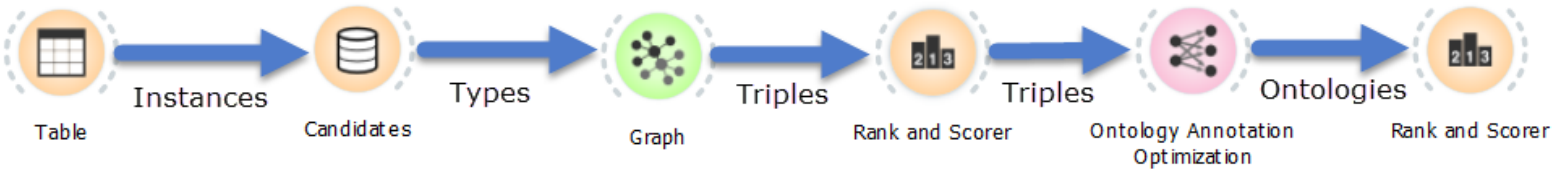


Figure 2: Fluid Diagram of the Schema annotation

Methodology: Pipe-Line 2

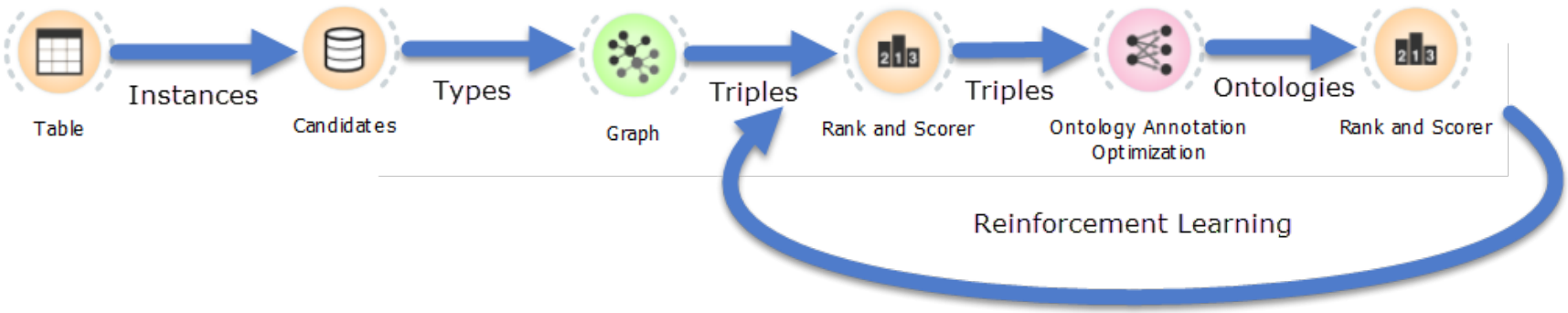
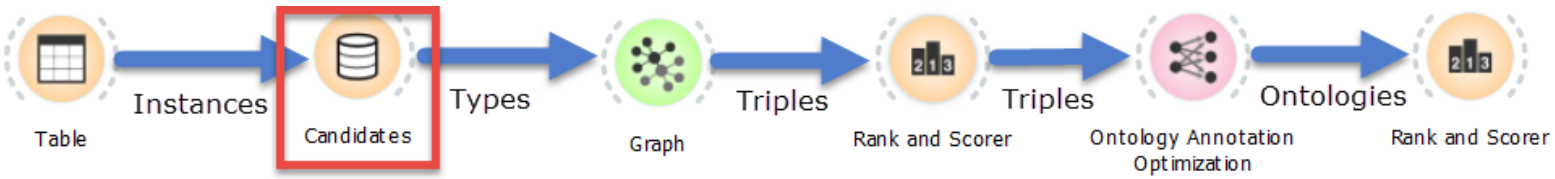


Figure 3: Fluid Diagram of the Schema annotation with the implementation of Reinforcement Learning

Methodology: Candidate - 1



Methodology: Candidate - 2

Definition (Type Candidate Maker)

datto SPARQL

Methodology: Candidate - 3

State of art

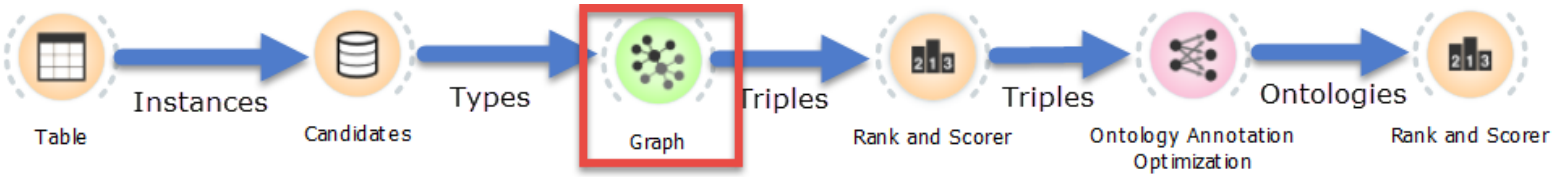
- SPARQL
- LOD TermPicker
- LOV
- Cod-start disambiguation TableMiner

Methodology: Candidate - 4

Scoring

- SPARQL
  - Counting Number of instances of each Type for each column

Methodology: Graph - 1



Definition (Predicate Candidate)

datto ABSTAT

Methodology: Graph - 3

State-of-the-Art

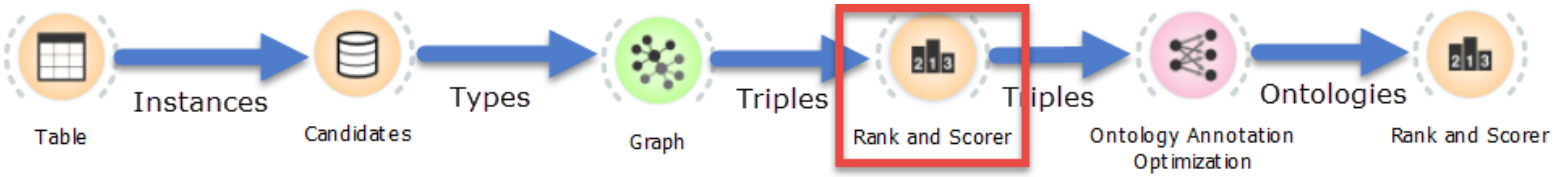
- ABSTAT
- Relation Enumeration/Literal column annoatation - Table Miner

Methodology: Graph - 4

Scoring

- Frequency ABSTAT

Methodology: Ranking Triples - 1



Methodology: Ranking Triples - 2

Definition (Triples Ranker)

a score is set given a condition

Methodology: Ranking Triples - 3

State-of-the-Art

- L2R (Learn to rank) TermPicker

Methodology: Ranking Triples - 4

Scoring

- subject indicator scorer
- ABSTAT on frequency usage

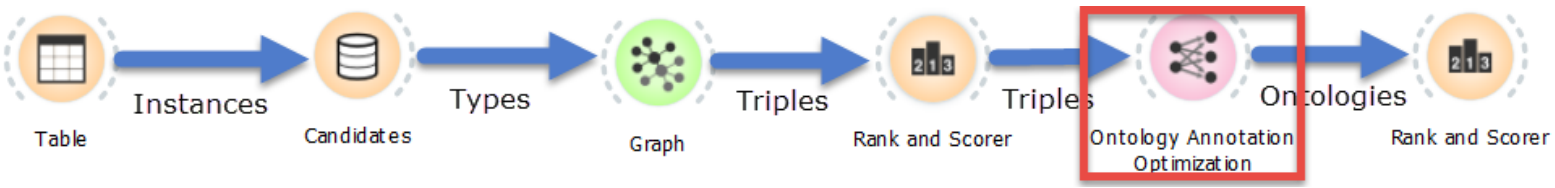


Figure 4: TableMiner design by Z.Zhang

Definition (Optimizer)

Find the path that maximize the path

Methodology: Making Ontologies

State-of-the-Art

Minimum Spanning Tree

Methodology: Ranking Ontologies

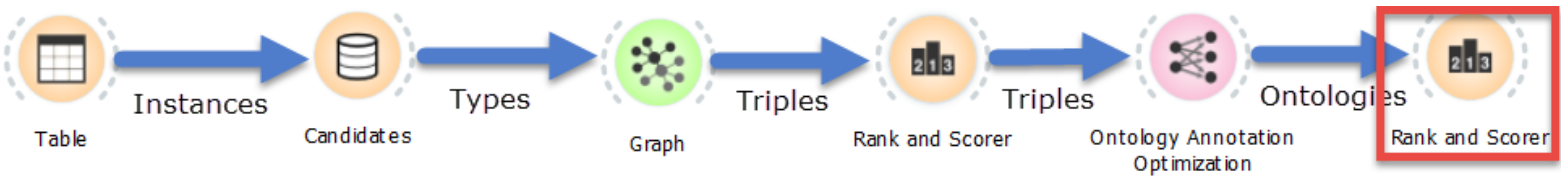


Figure 5: TableMiner design by Z.Zhang

Methodology: Ranking Ontologies

Definition (Ontologies Scorer)

scoring ontologies due to the mapping on other ontologies

Methodology: Ranking Ontologies

State-of-the-Art

- boh!

Methodology: Ranking Ontologies

Scoring

- the group that is more closer between the types of each possible ontology
- the number of ontologies that depends the result ontology

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Methodology: Result

Methodology: Result

Introduction:

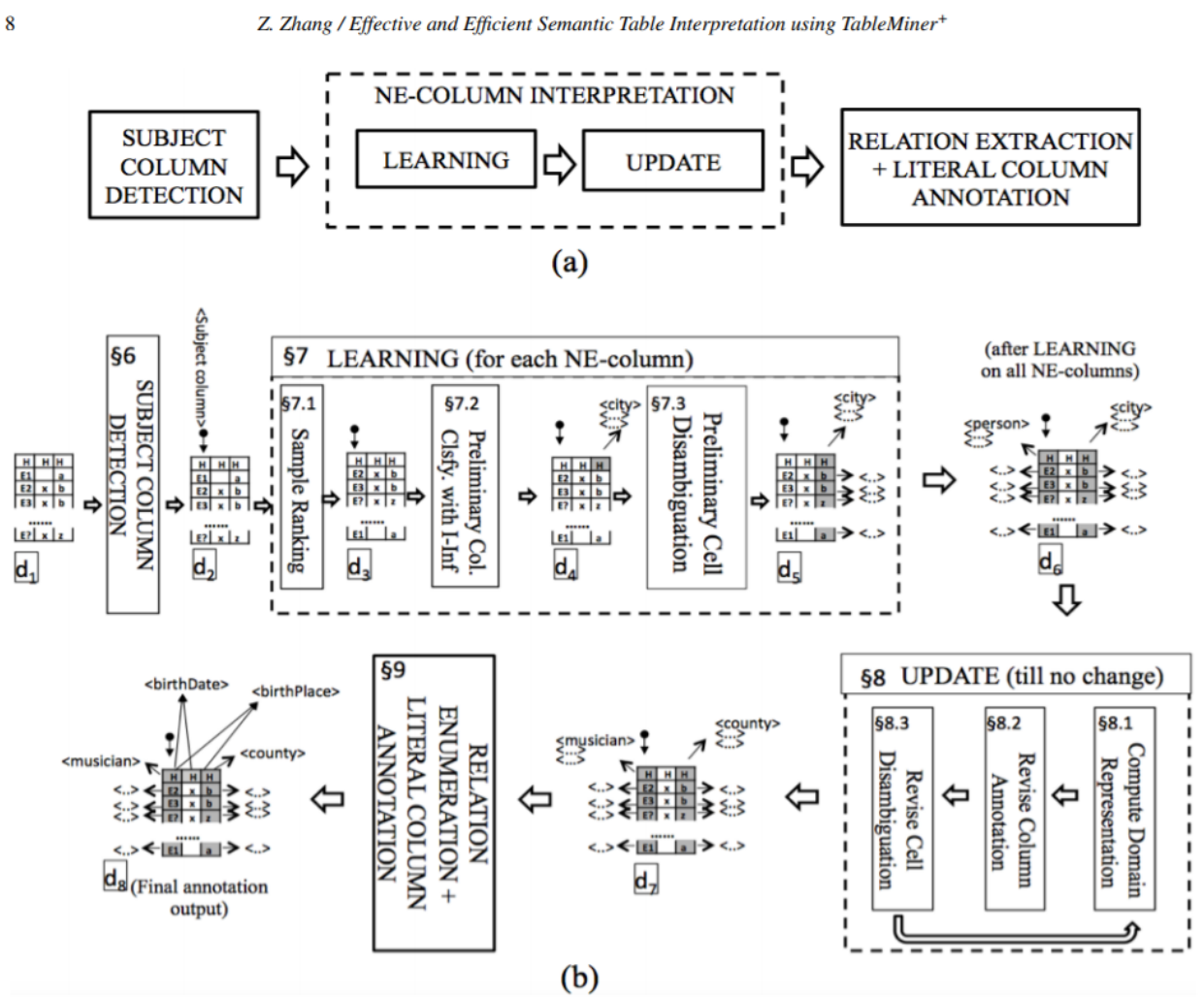


Fig. 2. The overview of TableMiner+. (a) a high-level architecture diagram; (b) detailed architecture with input/output. d - table data. Grey colour indicates annotated table elements. Angle brackets indicates annotations. Inside a table: H - header. E, a, b, x, z - content cells

Figure 6: TableMiner design by Z.Zhang

- Query Input
- Recommender of Vocabulary Term
  - Features for ranking
  - Learning to Ranking
- Query Output

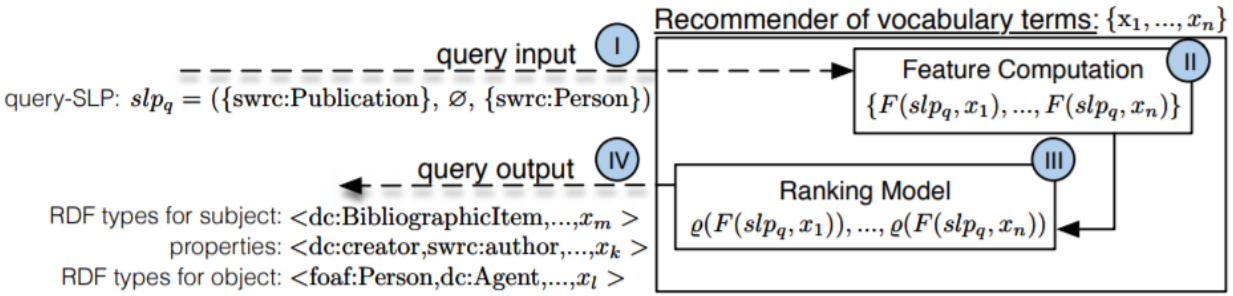


Figure 7: Schema of the process of the query

Dataset ABSTAT and ontologies

Conclusion

Reference

- SPARQL types
- LOD TermPicker
- LOV (others)
- ABSTAT
- TableMiner