Table Interpretation

University of Milan - Bicocca, Viale Sarca 336, Milan, Italy

Abstract.

1 The inputs of table interpretation: TABLE

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

Definition 2 (Rows and Columns). *Given an $n \times m$ table, let r_i denote the i-th row of the table, that is $r_i = \{(i, j) | 1 \leq j \leq n\}$ and c_j denote the j-th column ($c_j = \{(i, j) | 1 \leq i \leq m\}$). Let $\mathcal{R} = \{r_i | 1 \leq j \leq n\}$ and $\mathcal{C} = \{c_j | 1 \leq i \leq m\}$ be the set of all rows and columns of the table, respectively.*

Definition 3 (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 4 (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 5 (Header). *Given a header table $T_h = (T, h)$, we define $\mathcal{H} = h(\mathcal{C})$ as the header of table T.*

2 The inputs of table interpretation: ONTOLOGY and KG

Definition 6 (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.*

Definition 7 (Knowledge Graph). *Given an ontology $\mathcal{O} = (\mathcal{N}, \mathcal{P}, \mathcal{A})$, a Knowledge Graph $\mathcal{KG}$ [1] 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 $\psi : \mathcal{V} \rightarrow \mathcal{N}_o$, 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 $\phi : \mathcal{E} \rightarrow \mathcal{E}$, which maps an edge to a predicate type (e.g. dbr:The Matrix dbo:starring dbr:Keanu Reeves, dbr:The Matrix dbo:releaseDate "1999").*

3 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:
$\mathcal{KG}s = \{\mathcal{KG}_1, \mathcal{KG}_2, \ldots, \mathcal{KG}_k\}$
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

Definition 8 (Concept Matcher). *Given a knowledge base $\mathcal{KG}_x$, the Concept Matcher is a function $\theta_x : T \rightarrow \mathcal{N}_{cx} \cup \emptyset$:*

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

Definition 9 (Datatype Matcher). *Given a knowledge base $\mathcal{KG}_x$, the Datatype Matcher is a function $\theta_x : T \rightarrow \mathcal{N}_{dx} \cup \emptyset$:*

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

Definition 10 (Entity Matcher). *Given a knowledge base $\mathcal{KG}_x$, an Entity Matcher is a function $\chi_x : T \rightarrow \mathcal{Z}_x \cup \emptyset$:*

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

Lemma 1. *Given a knowledge base $\mathcal{KG}_x$ and an Entity Matcher is a function χ_x, a particular Concept Matcher is defined as:*

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

$$t_{ij} \in T \xrightarrow{\chi_x} z \in \mathcal{Z}_x \xrightarrow{\psi_x} c \in \mathcal{N}_{cx}$$

Definition 11 (Name Entity Identifier). *Given a knowledge base $\mathcal{KG}_x$, a Name Entity Identifier is function $\alpha_x : \mathcal{C} \rightarrow \{$ "Name Entity", "Literal"$\}$.*

Lemma 2. *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:*

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

a particular Name Entity Identifier can be defined as:

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

Definition 12 (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 $\sigma_x(\mathcal{T})_j = 0,\ \forall j \in \{j | \alpha_x(c_j) = 1\}$.*

Definition 13 (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 $\zeta_x : \mathcal{C} \rightarrow \mathcal{N}_x$.*

Lemma 3. *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:*

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

The function:

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

is a Semantic Column Annotator.

Definition 14 (Predicate Matcher). $\pi_x : \mathcal{C}^2 \rightarrow \mathcal{A}_x \cup \emptyset$ *with* $\pi_x(c_i, c_j) = \emptyset, \forall (i, j) | i = j$.

References

1. Baoxu Shi and Tim Weninger. Discriminative predicate path mining for fact checking in knowledge graphs. *Knowledge-Based Systems*, 104:123 – 133, 2016.