mantistable.disco.unimib.it

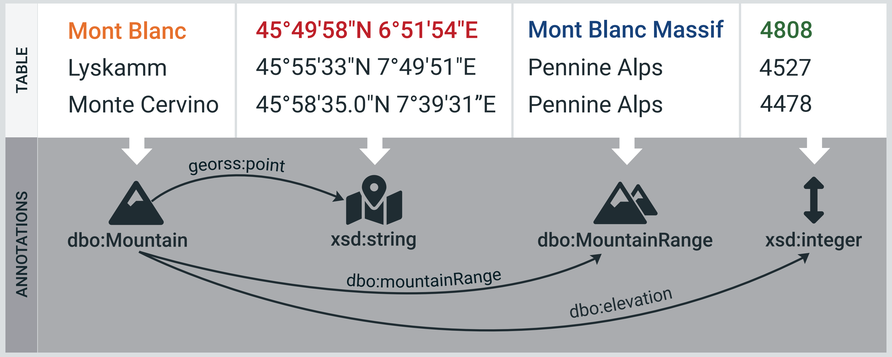
Semantic Table Interpretation using MANTISTABLE

Marco Cremaschi, *Anisa Rula, Alessandra Siano and Flavio De Paoli*
marco.cremaschi@unimib.it

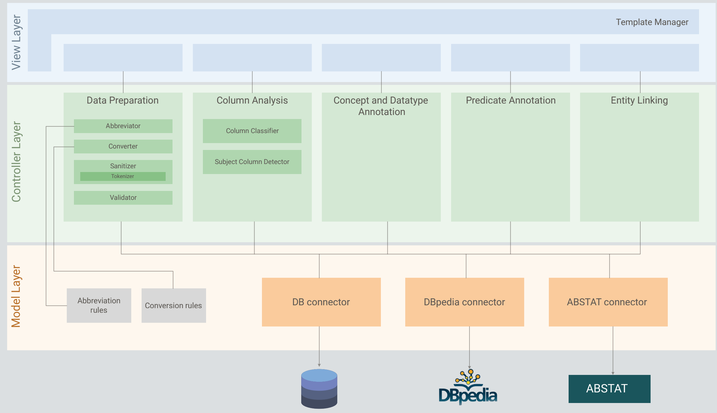
University of Milano - Bicocca
Milano, Italy

MantisTable is an open source **Semantic Table Interpretation** tool that automatically annotates, manages and makes accessible to humans and machines the **semantic of tables**.

SEMANTIC TABLE INTERPRETATION



MANTISTABLE ARCHITECTURE



MantisTable architecture is designed to be **modular**:

View Layer provides a **graphic user interface** to serve different types of tasks such as:

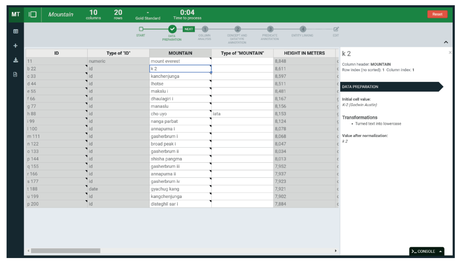
- storing and loading tables
- execution of the STI steps
- exploration of the annotated tables
- annotations editing

Controller Layer creates all the abstraction between the View layer and the Model layer and implements all the STI steps.

Model Layer considers mainly data access for communicating with an application's data sources such as DB connector or DBpedia connector.

MANTISTABLE PROCESS

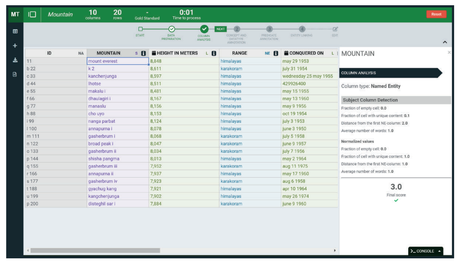
0 DATA PREPARATION



The **data** in the table are **cleaned** and **uniformed**

- Deletion of HTML tags and some characters
- Transformation of text into lowercase
- Deletion of text in brackets
- Explanation of acronyms and abbreviations using Oxford English Dictionary
- Normalisation of units of measurement applying regular expressions

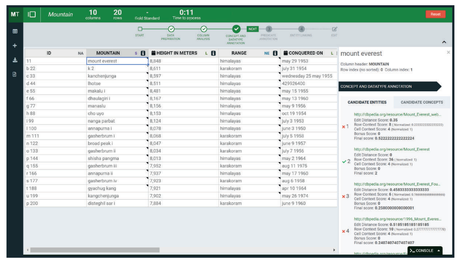
1 COLUMN ANALYSIS



Columns are classified as named-entity column (**NE-column**) or literal column (**L-column**) and a subject column (**S-column**) is detected

- Detection of **L-columns** by 16 regular expressions to identify regextype (e.g., geo coordinate, address, hex color code, URL)
- Detection of **S-column** considers different statistic features

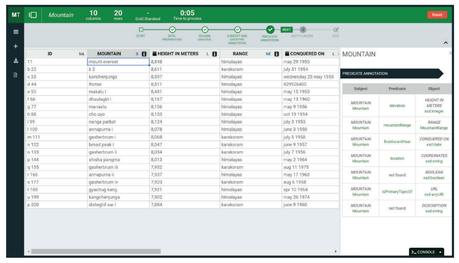
2 CONCEPT AND DATATYPE ANNOTATION



Column headers are mapped to **semantic elements** (*concepts* or *datatypes*) of a **Knowledge Graph**

- Retrieval of a set of **candidate entities** performing the **entity-linking** by searching the Knowledge Graph with the content of a cell. The entity with the highest **confident score** is used to annotate the cell
- Extraction of the ***rdf:type*** values for each winning entity. The most frequent type is used to annotate the column

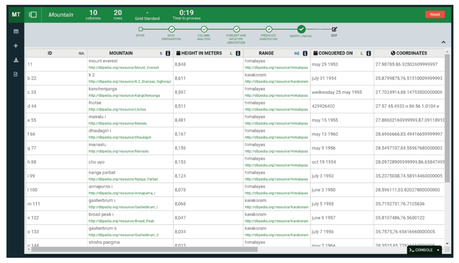
3 PREDICATE ANNOTATION



Relations (predicates) between the subject column and the other columns are identified

- The winning concept of the **S-column** are considered as the **subject** of the relationship and annotations of the other columns as **objects**
- The **Knowledge Graph** is searched for the subject and the object to collect possible predicates

4 ENTITY LINKING



The **content of cells** is mapped to entities in the **Knowledge Graph**

- Already discovered annotations are used to create a query for the **disambiguation of the cell content**
- If more than one entity is returned for a cell, the one with a smaller edit distance is taken