Semantic Table Interpretation using MantisTable

Marco Cremaschi[1], Anisa Rula[1,2], Alessandra Siano[1], and Flavio De Paoli[1]

[1] University of Milano - Bicocca, Italy
{marco.cremaschi,anisa.rula,flavio.depaoli}@unimib.it
a.siano2@campus.unimib.it
[2] Univeristy of Bonn, Germany
{rula}@cs.uni-bonn.de

Keywords: Knowledge Graph · Semantic Interpretation · Table Annotation

1 Introduction & Motivation

A vast amount of relevant structured data represented in tables are available on the Web. However, querying such data is difficult since they are incorporated in HTML web pages and are not easily query-able. Some approaches started to propose [4, 2] extraction, annotation and transformation of tabular data into machine-readable formats. The problem of annotating tables also known as *Semantic Table Interpretation (STI)* takes a relational table and a Knowledge Graph (KG) in input, and returns a semantically annotated table in output [1]. In this paper, we propose *MantisTable*[3], a web interface and an open source Semantic Table Interpretation tool that automatically annotates, manages and makes accessible to humans and machines the semantic of tables. Although STI contains several steps the key feature of our tool is the involvement of all the STI steps that run fully automatically.

2 Overview of MantisTable

Figure 1 shows the architecture of MantisTable which is designed to be modular:

View Layer provides a graphic user interface to serve different types of tasks such as storing and loading tables, exploration of the annotated tables which allow users to navigate all the executed steps by clicking on each phase and analyse the result, execution of the STI steps and the editing which allow users to understand what has been achieved and give them the opportunity to modify and enhance the results.

Controller Layer creates all the abstraction between the View layer and the Model layer and implements all the STI steps as follows:

Data Preparation cleans and normalizes values in the table. Transformations applied to tables include text normalization such as solve acronyms and abbreviations by applying regular expressions [3]. **Column Analysis** assigns types

[3] http://mantistable.disco.unimib.it

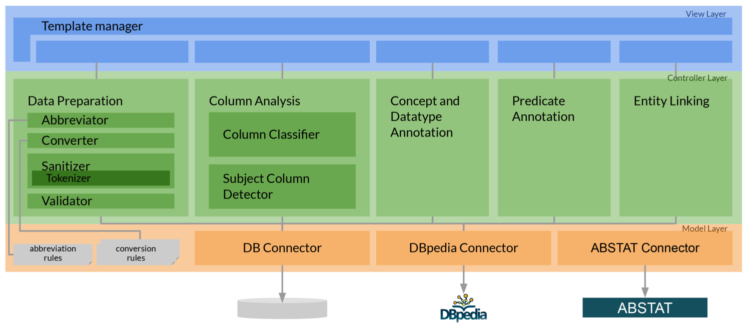

Fig. 1. Architecture of MantisTable tool.

to columns that are named entity (NE-column) or literal column (L-column), and then identify the subject column (S-column). To identify L-column candidates the tool considers 16 regular expressions that identify several Regextypes. If the number of occurrences of the most frequent Regextype in a column exceeds a given threshold, that column is annotated as L-column, otherwise, it is annotated as NE-column. To detect the S-column, the tool considers the NE-columns on which applies different statistic features (e.g. % of cells with unique content). **Concept and Datatype Annotation** identifies the mappings between columns headers and semantic elements (concepts or datatypes) in a KG. First, we perform the entity-linking by searching the KG with the content of a cell, to get a set of candidate entities and use the DICE similarity measure for text disambiguation. Second, the abstract and all concepts for each winning entity are retrieved from DBpedia. For each extracted concept, we count the occurrences in the abstract. For the Datatype Annotation we consider the L-columns and for the identification of datatypes, a Regextype is applied on the content of each column. **Predicate Annotation** finds relations, in the form of predicates, between the S-column and the object columns to set the overall meaning of the table. Further, the entities identified as subjects and objects are searched in the KG to identify the correct predicate. **Entity Linking** deals with mappings between the content of cells and entities in the KG.

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

References

1. Cremaschi, M., Rula, A., Siano, A., De Paoli, F.: MantisTable: a Tool for Creating Semantic Annotations on Tabular Data. In: 16th ESWC: Posters and Demos (2019)
2. Pham, M., Alse, S., Knoblock, C.A., Szekely, P.: Semantic labeling: a domain-independent approach. In: 15th ISWC (2016)
3. Ritze, D., Lehmberg, O., Bizer, C.: Matching HTML Tables to DBpedia. In: WIMS (2015)
4. Zhang, Z.: Effective and efficient semantic table interpretation using tableminer+. Semantic Web (2017)