MantisTable: an automatic approach for the Semantic Table Interpretation

Marco Cremaschi, Roberto Avogadro, and David Chieregato

University of Milan - Bicocca, Viale Sarca 336, 20126 Milan, Italy
marco.cremaschi@unimib.it
{r.avogadro,d.chieregato}@campus.unimib.it

1 Presentation of the system

1.1 State, purpose, general statement

On the Web we can find a vast amount of structured data, represented in tables that contain relevant information. Despite the huge corpus of such tables on different topics, they set limitations on artificial intelligence tasks, such as semantic search and query answering. This is the reason why some approaches started to propose extraction, annotation and transformation of tabular data into machine-readable formats. In particular, in the last years, there has been a ton of works on the annotation of tabular data, also known as *Semantic Table Interpretation (STI)*, which can be mainly classified as supervised (they exploit already annotated tables for training) [6,11,8,2] or unsupervised (they do not require training data) [14,9,1,7]; and as automatic [6,8,4] and semi-automatic [2]. Moreover, some approaches [3,14,9,1,7,13,12] focus mainly on the analysis of Web tables' context such as Web page title, table caption, or surrounding text, while others [6,11,8,4,10] address independent tables which can only rely on their own data. We identify some limits of the state-of-the-art approaches as follows: i) they adopt lexical comparisons for matching which ignore the contextual semantics; ii) they rely on metadata like column names and sometimes even external information like table descriptions, both of which are often unavailable in real world applications; iii) they use personalised Knowledge Graph (KG); iv) they perform only a few steps of STI. To overcome such limitations, we propose a comprehensive approach and a tool named *MantisTable*[1], which provides an unsupervised method to annotate independent tables, possibly without a header row or other external information. MantisTable takes a *well-formed and normalised* relational table (i.e. a table with headers and simple values, thus excluding nested and figure-like tables), and a KG which describes real-world entities in the domain of interest (i.e. a set of concepts, datatypes, predicates, entities, and the relations among them) in input, and returns a semantically annotated table in output. This process comprises different steps to semantically annotate tables, such as *semantic classification* of columns, which classifies a column either as a literal or as a named entity. Besides the approach, we propose

[1] mantistable.disco.unimib.it

MantisTable tool, a web interface and an open source Semantic Table Interpretation tool that automatically annotates, manages and makes the semantic of tables accessible to humans and machines. This tool is independent of any particular context. Additional built-in guidance functionalities help to avoid common pitfalls and create correct annotations. Although a STI contains several steps, as will be explained in the next section, the key feature of our approach is the involvement of all the STI steps that run fully automatically. This approach and tool were developed by one PhD student and two master's students.

1.2 Specific techniques used

The *MantisTable* approach implements STI steps through five phases:

0. **Data Preparation**, which aims to prepare the data inside the table;
1. **Column Analysis**, whose tasks are the semantic classification that assigns types to columns (NE-column or L-column), and the detection of the subject column (S-column);
2. **Concept and Datatype Annotation**, which deals with mappings between columns (or headers, if they are available) and semantic elements (concepts or datatypes) in a KG;
3. **Predicate Annotation**, whose task is to find relations, in the form of predicates, between the main column and the other columns to set the overall meaning of the table;
4. **Entity Linking**, which deals with mappings between cells and entities in a KG.

To describe each phase of the STI approach we consider Table 1[2], which lists video games with additional information, such as publisher, release date, etc.

Table 1. Table with a list of video games extracted from Round 1.

Title	Publisher	EU Release Date	AU Release Date	PEGI	ACB
Donkey Kong Country	Nintendo	2006-12-08	2006-12-07	7	G
Super Castlevania IV	Konami	2006-12-29	2006-12-29	3	PG
DoReMi Fantasy: Milon's DokiDoki Adventure (900 Wii Points)	Hudson Soft	2008-09-05	2008-09-05	3	G
...					

Data Preparation aims to clean and uniform data inside the table. Transformations applied to tables are as follows: deletion of HTML tags and some characters (i.e. " '), transformation of text into lowercase, deletion of text in brackets, resolution of acronyms and abbreviations, and normalisation of units of measurement. To decrypt acronyms and abbreviations, the Oxford English Dictionary[3] is used. The normalisation of units of measurement is performed by applying regular expressions, as described in [9]. MantisTable extends the original set of regular expressions to cover a complete set of units, which includes

[2] Round 1 table index: 11833461_1_3811022039809817402
[3] public.oed.com/how-to-use-the-oed/abbreviations/

area, currency, density, electric current, energy, flow rate, force, frequency, fuel efficiency, information unit, length, linear mass density, mass, numbers, population density, power, pressure, speed, temperature, time, torque, voltage and volume.

Column Analysis whose tasks are the *semantic classification* that assigns types to columns that are named entity (NE-column) or literal column (L-column), and the *detection of the subject column* (S-column). The first step of the Column Analysis phase is to identify good L-column candidates. To accomplish this task, we consider 16 regular expressions that identify several Regextypes (e.g. numbers, geo coordinate, address, hex color code, URL). 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. The second step deals with the *subject column detection* that takes into account the identified NE-columns. We can define the S-column as the main column of the table based on different statistic features, like Average Number of Words (aw) in each cell, Fraction of Empty Cells (emc) in the column, Fraction of Cells with Unique content (uc) and Distance from the First NE-column (df). These features are combined to compute the $subcol(c_j)$ score for each NE-column as follows:

$$subcol(c_j) = \frac{2uc_{norm}(c_j) + aw_{norm}(c_j) - emc_{norm}(c_j)}{\sqrt{df(c_j) + 1}} \quad (1)$$

The column with the highest score will be selected as the S-column for the considered table. The values of the features for the S-column detection related to the video games table (Table 1) are shown in Table 2. In this case the Title column is the S-column of the table (Table 3).

Table 2. Values of the features of the S-column detection for the video games table.

Feature	Title column	Publisher column
emc	0	0
uc	1	0.21
df	1	2
aw	1	0.37
final	3	0.57

Table 3. Table 1 after the Column Analysis phase.

S	**NE**	**L**	**L**	**L**	**NA**
Title	**Publisher**	**EU Release Date**	**AU Release Date**	**PEGI**	**ACB**
donkey kong country	nintendo	2006-12-08	2006-12-07	7	g
...					

Concept and Datatype Annotation deals with mappings between columns headers and semantic elements (concepts or datatypes) in a KG. In the first step of Concept Annotation, we perform the entity-linking on the cells in a subset of the rows of the table by searching the KG with the content of a cell $tx(i, j)$, to get a set of candidate entities $e_{i,j} \in E_{i,j} \in E$. We use the similarity between the content of the cell and the candidate entities to disambiguate the $tx(i, j)$. Given a candidate entity $e_{i,j}$, the similarity depends on two components: *entity context* and *entity name*. Entity context is a score that represents the similarity between the representation of the entity in the KG with the row

and the column elements to which the cell belongs to. Entity name is a score that represents the similarity between the name of the entity in the KG and the text in the cell which is under examinations. Entity context, EC, is calculated by computing a candidate entity $e_{i,j}$ with the cell's context $x_{i,j} \in X_{i,j}$ which considers header and row content: i) row content: is the concatenation of all the words in the cells in the same row j from every columns i, without considering the content of cell (i, j); ii) header content: is the concatenation of all the words of header $(0, j)$ plus the concatenation of all the synonyms (e.g. from Wordnet[4] or Oxford dictionary[5]).

We calculate the EC as follow:

$$EC(e_{i,j}) = |bow(abst.(e_{i,j})) \cap bow(row(i,j))| + |bow(abst.(e_{i,j})) \cap bow(header(i,j))| \quad (2)$$

Entity name EN is calculated by computing the edit distance (Levenshtein) between the labels (in different languages) of candidate entity $e_{i,j} \in E_{i,j}$ and the content of the cell $tx(i, j)$:

$$EN(e_{i,j}) = editDistance(tx(i,j), e_{i,j}) \quad (3)$$

The final objective is to identify the entity with the highest confidence score ECF, which will then be used for annotating the cell. The confidence score ECF is computed as follows:

$$ECF(e_{i,j}) = bonus(e_{i,j}) + econtext(e_{i,j}) - ename(e_{i,j}) * 2 \quad (4)$$

The score $bonus(e_{i,j})$ in the Formula 4 is used to rank the entities most related to the content of the cell by considering the presence of the tokens of the text in the cell, within the entity labels and the entity abstract (Formula 5).

$$bonus(e_{i,j}) = |bow(tx(i,j)) \cap bow(e_{i,j})| + |bow(tx(i,j)) \cap bow(abstract(e_{i,j}))| \quad (5)$$

The entity $e_{i,j}$ with the highest score will be selected as the winning entity to associate to $tx(i, j)$. For each cell $tx(i, j)$ a set of candidate entities $E_{i,j}$ is extracted from the KG, through the query shown in Listing 1.1. The query searches the entities considering both the entire content of the cell and the individual words. In addition, we search the descriptions associated with the entities according to the synonyms of the header. Values in the header are assumed to be nouns, thus the respective synonyms are extracted from WordNet, which is a semantic-lexical database of the English language, and from the thesaurus of Oxford dictionary. For instance, considering the example in Table 1, the approach searches the KG with the synonyms of Title, which is the header of the S-column, which results in *name, subject, caption, publication, etc.*[6]. The maximum number of results per query is set at 10; this number has been defined empirically with several tests which gave evidence that the correct result is mostly within the first 5 results.

[4] wordnet.princeton.edu
[5] oed.com
[6] www.lexico.com/en/synonym/title

Listing 1.1. SPARQL query to retrieve a set of candidate entities for a text in a cell.

```
SELECT DISTINCT (str(?s) as ?s) (str(?abstract) as ?abstract)
WHERE {
{
    ?s dbo:abstract ?abstract .
    ?s a ?type .
    ?s rdfs:label ?label .
    ?label <bif:contains> 'donkey AND kong AND country' .
    ?abstract <bif:contains> '("title" OR "name" OR "subject" OR [synonyms])'
  }
[..]
}
ORDER BY ASC(strlen(?label))
LIMIT 10
```

In this example the row and header content are: i) row content: donkey kong country nintendo 2006-12-08 2006-12-07 7 G; ii) header content: the synonyms of the header "title" are "name, subject, publication, ...".

Applying the Formulas 2,3,4, the EC, EN and ECF are calculated for candidate entitites (see Listing 1.2).

Listing 1.2. List of entities with entity context score, entity name score and confidence score.

```
"dbr:Donkey_Kong_Country"
  score EC 0.61, score EN 0, score BONUS 1, score ECF 2.11 # winning entity
"dbr:Donkey_Kong_Country_Returns"
  score EC 0.57, score EN 0.29, score BONUS 1, score ECF 1.46
"dbr:Donkey_Kong_Country:_Tropical_Freeze"
  score EC 0.30, score EN 0.45, score BONUS 1, score ECF 0.49
[..]
```

In this case the winning entity is `dbr:Donkey_Kong_Country`[7].

In the second step of Concept Annotation, a set of concepts $CO_{i,j} \in CO$, associated with the winning entities $e_{i,j}$ and identified in the previous step, are obtained. This set will then be used to identify a concept to be associated with the column. In particular, for each winning entity, all the `rdf:type` values are extracted. Afterwards, for each type, labels (`rdfs:label`) are retrieved. For each label the occurrence rate is calculated between all values of the `rdf:type`. The concept $CO_{i,j}$ with the highest score will be selected as the winning concept to be associated with the column. The final result from the phase is the one shown in Table 4.

Table 4. Global frequency values for the Title column.

Type	Global frequency	# cells
VideoGame	36	27
TelevisionShow	11	2
Software	36	27
Device	1	1
Work	36	27

The hierarchy of the concepts is `Work > Software > VideoGame, Device, TelevisionShow`. The occurrences of `Work` and `Software` are added to the occurrences of `VideoGame`. The minimal concept is used to annotate the column.

For the Datatype Annotation, the results of the Column Analysis are taken into consideration. In that phase, a column gets associated with a specific Regextype. To identify the correct Datatype, a mapping between the Regextype and

[7] dbpedia.org/resource/Donkey_Kong_Country

the Datatype was created. Table 5 shows the example in Table 1 with final columns annotations.

Table 5. Table 1 with annotations.

dbo:VideoGame	**dbo:Company**	**xsd:date**	**xsd:date**	**xsd:integer**	**NA**
Title	**Publisher**	**EU Release Date**	**AU Release Date**	**PEGI**	**ACB**
donkey kong country	nintendo	2006-12-08	2006-12-07	7	g
...					

Predicate Annotation, whose task is to find relations in the form of predicates, between the Subject column and the other columns, to set the overall meaning of the table. MantisTable approach considers the winning concept of the S-column as the subject of the relationship, and the remaining columns as objects. The entities identified as subjects and objects are further searched in the KG. In order to identify the correct predicate, we compare the content of the column and the candidate predicates. Given a candidate predicate, the confidence score depends on two components: the *predicate context* and the *predicate frequency*. Predicate context PC is a score that represents the similarity between the representation of the predicate from the KG and the representation of the NE-column. PC is calculated by comparing a candidate predicate p_j with the column context $x_j \in X_j$, which is further divided into in-table context (column header and column content) and out-table context (additional information extracted from web). We calculate the overlap between the representation of the candidate property p_j and the representation of each context x_j using the Dice similarity, computed as follows:

$$PC(p_j) = \frac{2 \cdot \sum_{w \in bowset(p_j) \bigcap bowset(x_j)} (freq(w, bow(p_j)) + freq(w, bow(x_j)))}{|bow(p_j)| + |bow(x_j)|} \quad (6)$$

PF refers to the ratio between the candidate predicate frequency and the sum of all candidate predicate frequencies. The selection of the predicate is computed using predicate confidence score PCF:

$$PCF(p_j) = pcontext(p_j) + pfreq(p_j) \quad (7)$$

The MantisTable approach can also identify the predicates between the subject column (S-column) and a literal column (L-column). We execute a query that, given the concept co_j of the S-column and the values of the L-column, finds all predicates where the subject is co_j and the objects are values. In contrast to the previous method (for the NE-column), we consider the synonyms of the headers (if present) in order to increase the number of predicate candidates. The scores described above are used to identify the winning predicate. For columns that contain numeric values, often difficult to annotate, we extended the idea proposed in [5], which applies a hierarchical clustering algorithm on a reference KG to build a Background Knowledge Graph (BKG). This BKG contains information about "numerical representative" of "contexts", i.e. predicates and their shared domains (subject concept). More specifically, given two numerical sets,

one from the table, and one in the BKG, we analyse and compare the distribution of numerical values through the distance of Kolmogorov - Smirnov. The application of this technique returns top-k candidate predicates with related concepts. In order to identify among the top-k the most suitable predicate for the annotation of the numerical column, it is possible to exploit the annotations obtained in the previous phases. Starting from the annotation co_j of the S-column, we propose a technique supporting the rearrangement of the top-k results on the basis of their relevance with respect to co_j to obtain the most probable predicate candidates for the numerical columns.

Entity Linking deals with mappings between the content of cells and entities in the KG. The annotations obtained in the previous steps are used to create a query for the disambiguation of the cell contents. However, the use of the winning class for filtering results was too stringent. This is because, from a series of experiments, it was noted that not all elements within a column have the same values for the `rdf:type` property due to some inconsistency of DBpedia. For this reason, to increase the number of results, we have chosen to consider only the entities for which, within the values of `rdf:type`, it is possible to find the label of the winning class. If more than one entity is returned for a cell, the one with a smaller edit distance (i.e. Wagner-Fischer distance) is taken.

2 Link to the system

As described above, the MantisTable approach has been integrated into a web application developed with Python and the Django. A MongoDB database acts as table and KG repository. The code is freely available through a Git repository[8]. In order to achieve the scalability of the application, and therefore improve efficiency, MantisTable has been installed in a Docker container to achieve parallelisation at the application level and to facilitate the deployment on servers. The management of resources is performed by using Task Queues (i.e. Celery Workers[9]). The five phases of the STI have been modularly implemented, allowing an easy replacement or extension by other developers.

2.1 Adaptations made for the evaluation

To participate in the challenge we made some changes as follows: i) MantisTable was originally developed to support the JSON format for loading tables and for exporting results. In order to take part in the challenge, we developed a new parser for managing the CSV tables. During this phase, we encountered several problems in the management of different characters encoding; ii) since target columns (columns to be annotated) were provided during the challenge, we disabled most of the Column Analysis phase (i.e. subject column detection); iii) our solution was made to identify just one correct Class per column, so we made a new export script with different criteria for the selection of concepts, also considering the hierarchy of these in the KG.

[8] bitbucket.org/disco_unimib/mantistable-tool.py
[9] docs.celeryproject.org/en/latest/userguide/workers.html

3 Results

In this section, the MantisTable approach's results in Round 1 and Round 2 of the challenge will be discussed. In the first round MantisTable achieved the results in Table 6. The results are good, in particular in the CEA task. Regarding the CTA and CPA results, it is necessary to view the dataset to understand which are the incorrect annotations. In the second round MantisTable achieved the results in Table 7.

Table 6. Results of Round 1.

TASK	F1-Score	Precision
CTA	0.929	0.929
CEA	1.0	1.0
CPA	0.965	0.991

Table 7. Results of Round 2.

TASK	AH-Score	AP-Score
CTA	1.049	0.247
CEA	0.614	0.673
CPA	0.460	0.544

We particularly focused on the CTA task because it's crucial to our algorithm for the other steps; we use the results of the Concept Annotation to filter the results in the CEA and CPA tasks. The main issues faced in this task were: i) tables with few rows (1-2) are sometimes difficult to be linked; this may be addressed by trying to integrate some other data sources (maybe considering Wikidata) for the entity linking; ii) wrong annotations with the new metric have an high incidence, some tuning of the thresholds regarding the scores obtained during the Concept Annotation phase may provide better quality annotations; iii) some target columns are very complex to annotate because the cell contents cannot be directly linked to entities in the KG, so we decided to exclude the columns that our tool identified as L-columns; iv) tables about people with only surnames are frequently linked to homonym entities, some specific solutions to expand the context in this kind of situation will be adopted. About the CEA task, probably between parsing and cleaning the data some rows were misplaced in different indexes. We didn't have much time to look into it but we are sure that we could have done a better job in this task using other resources for disambiguation (e.g. DBpedia or Wikidata). In the next round, it will be our main goal to focus on CPA and CEA tasks.

4 Conclusions and General comments

Unlike the state of the art approaches, MantisTable i) provides a comprehensive solution to support all annotations steps; ii) provides an unsupervised method to annotate independent tables; iii) generates context for disambiguation; iv) provides a tool to support STI workflow and a tool to support the evaluation by providing validation indicators which are both publicly available. In relation to the results obtained in round two, we are making a series of adjustments, such as the use of external resources to better the content's disambiguation within each cell during the Concept Annotation phase. During the analysis of the dataset, we noticed how some columns have non-annotable elements (e.g. numeric values or codes) or the presence of tables with an incorrect structure (e.g. presence of multiple headers, different number of columns in the same table). We have created a list with these tables in order to improve the dataset.

References

1. Deng, D., Jiang, Y., Li, G., Li, J., Yu, C.: Scalable column concept determination for web tables using large knowledge bases. Proc. VLDB Endow. **6**(13), 1606–1617 (Aug 2013)
2. Knoblock, C.A., Szekely, P., Ambite, J.L., Goel, A., Gupta, S., Lerman, K., Muslea, M., Taheriyan, M., Mallick, P.: Semi-automatically Mapping Structured Sources into the Semantic Web, pp. 375–390. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
3. Limaye, G., Sarawagi, S., Chakrabarti, S.: Annotating and searching web tables using entities, types and relationships. Proc. VLDB Endow. **3**(1-2), 1338–1347 (Sep 2010)
4. Mulwad, V., Finin, T., Joshi, A.: Semantic message passing for generating linked data from tables. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) The Semantic Web – ISWC 2013. pp. 363–378. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
5. Neumaier, S., Umbrich, J., Parreira, J.X., Polleres, A.: Multi-level semantic labelling of numerical values. In: Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y. (eds.) The Semantic Web – ISWC 2016. pp. 428–445. Springer International Publishing, Cham (2016)
6. Pham, M., Alse, S., Knoblock, C.A., Szekely, P.: Semantic Labeling: A Domain-Independent Approach, pp. 446–462. Springer International Publishing, Cham (2016)
7. Quercini, G., Reynaud, C.: Entity discovery and annotation in tables. In: Proceedings of the 16th International Conference on Extending Database Technology. pp. 693–704. EDBT '13, ACM, New York, NY, USA (2013)
8. Ramnandan, S., Mittal, A., Knoblock, C.A., Szekely, P.: Assigning Semantic Labels to Data Sources, pp. 403–417. Springer International Publishing, Cham (2015)
9. Ritze, D., Lehmberg, O., Bizer, C.: Matching html tables to dbpedia. In: Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics. pp. 10:1–10:6. WIMS '15, ACM, New York, NY, USA (2015)
10. Syed, Z., Finin, T., Mulwad, V., Joshi, A.: Exploiting a web of semantic data for interpreting tables. In: Proceedings of the Second Web Science Conference. vol. 5 (2010)
11. Taheriyan, M., Knoblock, C.A., Szekely, P., Ambite, J.L.: Learning the semantics of structured data sources. Web Semantics: Science, Services and Agents on the World Wide Web **37–38**, 152 – 169 (2016)
12. Venetis, P., Halevy, A., Madhavan, J., Paşca, M., Shen, W., Wu, F., Miao, G., Wu, C.: Recovering semantics of tables on the web. Proc. VLDB Endow. **4**(9), 528–538 (Jun 2011)
13. Wang, J., Wang, H., Wang, Z., Zhu, K.Q.: Understanding tables on the web. In: Proceedings of the 31st International Conference on Conceptual Modeling. pp. 141–155. ER'12, Springer-Verlag, Berlin, Heidelberg (2012)
14. Zhang, Z.: Effective and efficient semantic table interpretation using tableminer+. Semantic Web **8**(6), 921–957 (2017)