KEPLER-ASI : KEPLER as A Semantic Interpreter

Wiem Baazouzi[1], Marouen Kachroudi[2], and Sami Faiz[3]

[1] Laboratoire de Recherche en génie logIciel, Application distribuées , systèmes Décisionnels et Imagerie intelligente,National School of Computer Science, Tunis
wiem.baazouzi@ensi-uma.tn
[2] Université de Tunis El Manar, Faculté des Sciences de Tunis, Informatique Programmation Algorithmique et Heuristique, LR11ES14, 2092, Tunis, Tunisie
marouen.kachroudi@fst.rnu.tn
[3] Institut Superieur Des Arts Multimedias De Manouba, UR Téledétection Et Systèmes D'informations A Référence Spatiale
sami.faiz@insat.rnu.tn

Abstract. This paper presents our system KEPLER-ASI, for the Semantic Web on Tabular Data Challenge to Knowledge Graph Correspondence (SemTab 2020).KEPLER-ASI analyze tabular data and detect the correct matches in Wikidata, where data and values are annotated with a unique tag. Indeed, this task is difficult for machines to identify the right meaning of a given annotation. KEPLER-ASI uses the SPARQL query to semantically annotate tables in Knowledge Graphs (KG), in order to solve the critical problems of the matching tasks, namely, CTA columns annotation.

Keywords: Tabular Data - Knowledge Graph - KEPLER-ASI - SPARQL - Semantic Web Challenge

1 Introduction

The World Wide Web contains vast quantities of textual information in several forms: unstructured text, template-based semi-structured Web pages (which present data in key-value pairs and lists), and obviously tables. Methods which aim to extract information from these resources to convert them into a structured form have been the subject of several works. As an observation, it is obvious that there is a lack of understanding for the semantic structure which can hamper the process of data analysis. Gaining this semantic understanding will therefore be very useful for data integration, data cleaning, data mining, machine learning, and knowledge discovery tasks. For example, understanding data can help assess the appropriate types of transformation on it.

Tabular data is routinely transferred on the Web in a variety of formats. Most of these data sets are available in tabular formats (*e.g.*, CSV, Excel). The main reason of this format popularity is simplicity: many common office tools

(*e.g.*, Excel) are available to facilitate their generation and exploitation. Tables on the Web are the source of a highly valuable data. The addition of semantic information to Web tables may enhance a wide range of applications, such as Web search, Question Answering, and Knowledge Base (KB) building.

Researchers have largely different problems about the data when they extract tabular data from the Web, such as learning with limited labeled data, defining (or avoiding defining) ontologies, making use of prior knowledge, and scaling solutions on the Web. This task is often difficult in practice due to metadata (*e.g.*, table and column names) being missing, incomplete or ambiguous.

In recent years, we have identified several works that can be mainly classified as supervised (in the form of annotated tables to carry out the learning task) [1–5] or unsupervised (tables whose data is not dedicated to learning) [6, 5]. To solve these problems, we propose a global approach named KEPLER-ASI, which addresses the challenge of matching tabular data to knowledge graphs.This method is based on previous work, which deals with ontology alignment. [7–9].

In this challenge, the input is a CSV file, but three different challenges had to be met :

1. **CTA :** A type of the Wikidata ontology had to be assigned a class KG to a column (Column-Type Annotation).
2. **CEA :** A wikidata entity had to be matched to the different cells (Cell-Entity Annotation).
3. **CPA :** A KG property had to be assigned to the relationship between two columns (Columns Property Annotation).

Data annotation is a fundamental process in tabular data analysis, it allows to infer the meaning of other information. Then deduce the meaning of a tabular knowledge graph. The data we used was based on Wikidata. We would like to mention, in a more general context, that Wikidata is made up of several types of documents, which obey the triples format : subject ($\mathcal{S}$), a predicate ($\mathcal{P}$) and an object ($\mathcal{O}$)

Indeed, Cell Entity Annotation (CEA) matches a cell to a KG entity. At this level, we have to annotate each individual element of the subject ($\mathcal{S}$) and the object ($\mathcal{O}$). Column Property Annotation (CPA) assigns a KG property to the relationship between two columns. The task is to find out which property of the two columns are connected to Wikidata. Column Type Annotation (CTA) assigns connected semantic type to a column. This work means another topic that can be described by including tags corresponding to the topic in Wikidata in common.

2 Knowledge Graph & Tabular Data

2.1 Tabular Data

S is a two-dimensional tabular structure made up of an ordered set of N rows and M columns (Fig 1). n_i is a row of the table (i = 1 ... N), m_j is a column of

the table (j = 1 ... M). The intersection between a row n_i and a column m_j is $c_{i,j}$, which is a value of the cell $S_{i,j}$. The table contents can have different types (string, date, float, number, etc.).

- Target Table (S): M × N.
- Subject Cell: $S_{(i,0)}$ (i = 1, 2 ... N).
- Object Cell: $S_{(i,j)}$ (i = 1, 2 ... M),(j = 1, 2 ... N).

$$
\begin{array}{c}
 \\
Row_1 \\
 \\
 \\
Row_j \\
 \\
 \\
Row_M
\end{array}
\begin{array}{c}
\begin{array}{cccccc}
\mathrm{Col}_0 & & \mathrm{Col}_i & & \mathrm{Col}_N
\end{array} \\
\left(\begin{array}{ccccc}
S_{1,0} & \dots & \dots & \dots & S_{1,N} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
S_{j,0} & \dots & S_{j,i} & \dots & S_{j,N} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
S_{M,0} & \dots & \dots & \dots & S_{M,N}
\end{array}\right)
\end{array}
$$

Fig. 1. Target Table

2.2 Knowledge Graph

Knowledge Graphs have been in the focus of research since 2012, resulting in a wide variety of published descriptions and definitions. The lack of a common core, a fact that is also indicated by Paulheim [10] in 2015. Paulheim listed in his survey of Knowledge Graph refinement, the minimum set of characteristics that must be present to distinguish knowledge graphs from other knowledge collections, which basically restricts the term to any graph based knowledge representation. In the online reviewing [10], authors agreed that a more precise definition was hard to find at that point. This statement points out the demand for closer investigation and deeper reflection in this area.

Farber et *al.* defined a Knowledge Graph as an Resource Description Framework (RDF) graph and stated that the term KG was coined by Google to describe any graph-based knowledge base (KB) [11]. Although this definition is the only formal one, it contradicts with more general definitions as it explicitly requires the RDF data model.

3 System Description

The proposed system solve only CTA task. Our system consists of three phases as flagged by Figure 2. Although there are a number of methods available, several ideas have been tried, but the most effective according to our objective was

the idea used in the Mantistable approach [5, 12], we have indeed adopted the following phases:

Phase 1 Data preparation: This phase is used to prepare the data inside the table.

Phase 2 Column Analysis: In this phase, we determined the semantic classification of the columns to determine Named-Entity column (NE-column), Literal column (L-column) or Subject column (S-column).

Column-Type Annotation (CTA) which deals with mappings between columns and semantic elements in a knowledge graph KG by using a SPARQL query. In the next section, you will find detailed information about each phase.

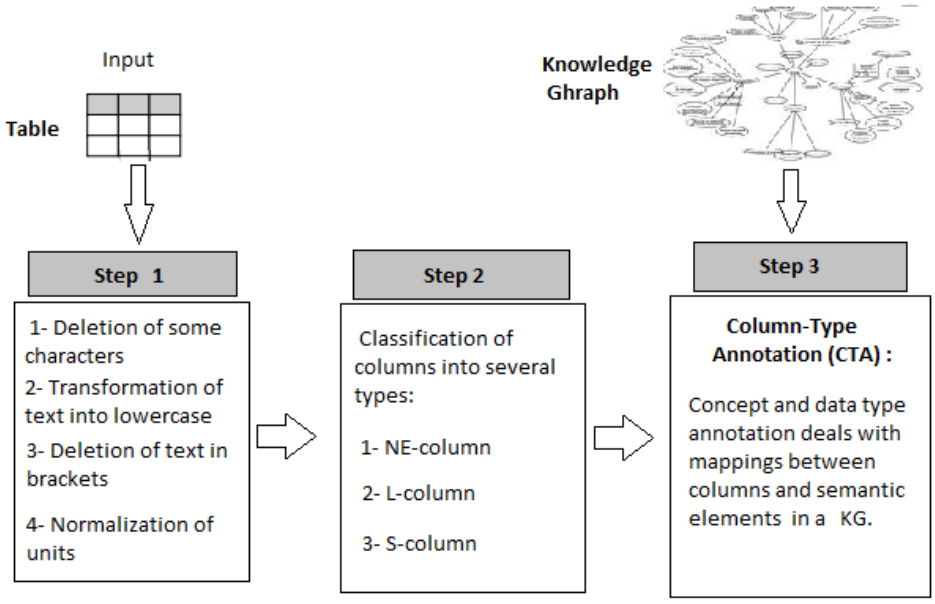

Fig. 2. Overview of our approach

3.1 Data Preparation

Data preparation aims to clean and standardize the data within the table. The transformations applied to tables are as follows:

1. The deletion of certain characters : For each of the cell values, we first clean them by retaining only the part that comes before a '(' or '[' and by removing all '
2. The transformation of text into lowercase, deletion of text in brackets, resolution of acronyms and abbreviations, and normalisation of units of measurement to decipher acronyms and abbreviations.
3. The normalization of the units of measurement is performed by applying regular expression treatments, as described in [13].

The use of regular expressions allows to devour a complete set of units, which includes area, currency, density, electric current, energy, flow, strength, frequency, energy efficiency, unit of information, length, linear mass density, mass, numbers, population density, power, pressure, speed, temperature, time, torque, voltage and volume.

3.2 Column Analysis

In this task we have classified the columns into several types with columns named entity (*NE-column*) or literal column (*L-column*) and detected of the subject column (*S-column*).

To accomplish this task, we consider 16 regular expressions that identify multiple Regextypes (for example, numbers, geographic coordinates, address, hexadecimal color code, URL).

To accomplish this task, we consider 16 regular expressions that identify multiple Regextypes (eg, numbers, geographic coordinates, address, hexadecimal color code, URL). Then, we set a threshold (equal to 0.7), if the number of occurrences of the Regextype in a column (for example an address) is the most frequent and exceeding this threshold, then this column is annotated as Column L, otherwise, it is annotated as **NE-column**. After the detection of this column (**L-column** or **NE-column**), we identified the subject column S-column.

Finaly to define the **S-column** as the main column of the table according to different statistical characteristics, namely [5] :

- **aw:** the average number of words in each cell.
- **emc:** the fraction of empty cells in the column.
- **uc:** the fraction of cells with unique content.
- **df:** the distance of the first NE-column.

These characteristics are combined to calculate the sub-column score (c_j) for each NE-column as follows [5] :

$$subcol(c_j) = \frac{2uc_norm(c_j) + aw_norm(c_j) - emc_norm(c_j)}{\sqrt{df(c_j) + 1}} \quad (1)$$

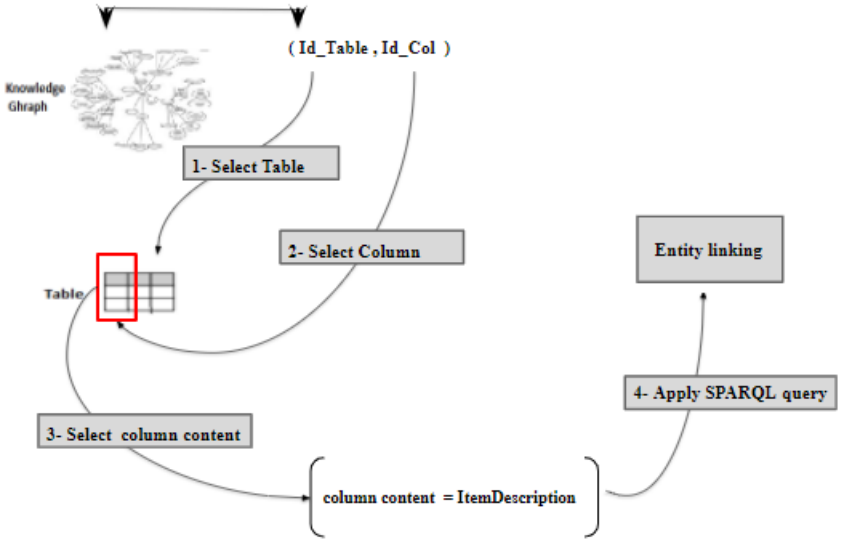

Fig. 3. Column-Type Annotation (CTA) steps

3.3 Column-Type Annotation (CTA)

Concept and data type annotation deals with mappings between columns and semantic elements (concepts or data types) in a KG. Figure (Fig 3) shows the four stages of our architecture for CTA. In the first concept annotation step, we started with an entity link found in KG with a column from a table, from common CSV files, then in the second step we fetched the contents of a column like ItemDescription in our SPARQL query to query wikidata, in order to find the caption of the winning class. The ItemDescription **"%s"** (Listing 1.1) on a Wikidata entry is a short phrase designed to disambiguate items with the same or similar labels. A description does not need to be unique; multiple items can have the same description, however no two items can have both the same label and the same description. If multiple entities were returned for a cell, the one with the number of occurrences was taken. For Correspondence of columns with Knowledge KG entities, All the inferred column types were taken into account using a simple SPARQL query:

Listing 1.1. SPARQL query to retrieve a set of entities eligible for the content of a column.

```
{
    SELECT ?itemLabel ?class
    WHERE {
                ?item    ?itemDescription "%s"@en .
                ?item wdt:P31 ?class
            }
}
```

4 Evaluation results

In this section, the results of the KEPLER-ASI approach during Rounds 1, 2, 3 and 4 of the challenge are presented. F1-Score and Precision values are listed in Table 1 for the CTA task.

Table 1. Results of the challenge

		AF1	APrecision
Round 1	**CTA**	-	-
	CPA	-	-
	CEA	-	-
Round 2	**CTA**	0.293	0.789
	CPA	-	-
	CEA	-	-
Round 3	**CTA**	0.381	0.701
	CPA	-	-
	CEA	-	-
Round 4	**CTA**	0.253	0.676
	CPA	-	-
	CEA	-	-

The values obtained in the 3 rounds are encouraging in relation to the volumes of data and the limits of the machines. This means that there are ways to investigate in terms of new technologies that can allow us to get around this kind of problem.

5 Conclusion Future Work

To sum up, we have developed a simple approach for automatic table annotation. While there are several techniques available, we have chosen a simpler approach. Our main effort was in the CTA task and how to proceed using a content-based SPARQL query, but our approach misses the preprocessing phase

and data correction. Several techniques were tried during preprocessing, but the most effective was spell checking. This pre-treatment may be improved to increase the precision values. We try to improve on this weak point of our approach in future work. We also have other problems, one run may take 12 hours due to the limitation of our machines. The execution of our system KEPLER-ASI requires a lot of instruction of reading and writing, to consult Wikidata. This requires a great resource in terms of RAM and processor, to improve the matching processes. We plan to investigate this lead in the near future to identify which resource needs to be further improved. So for a large data set running a job locally was not possible in our case.

In this article, we presented our contribution to the SemTab2020 challenge, KEPLER-ASI. We tackled a posed task, the CTA. We base our solution only on SPARQL queries using the cell contents as a description of a given item. Our main effort was in using the cell contents as a description of a given item. KEPLER-ASI is a simple approach but we will improve our preprocessing phase for two main purposes: First, we will apply a method to correct spelling mistakes and other typos in the source data. Second, we'll determine the data type of each column. Although the system distinguishes more types of data: OBJECT, DATE, STRING and NUMBER. Finally, due to the small size of the used machines during in the different evaluation phases (Intet (R) Core (TM) i5-7200U CPU @ 2.50GHZ, 2701 MHz, 2 cores 4 processors, with 8 GB of RAM), we will try to develop our system by integrating new data processing techniques. Eventually, the idea of moving to a data representation using indexes would be a good track to investigate in order to master the search space. In addition, the processing parallelism will allow us to circumvent the problem of the data size which is the major gap for our current machines.

References

1. Pham, M., Alse, S., Knoblock, C.A., Szekely, P.: Semantic labeling: a domain-independent approach. In: International Semantic Web Conference, Springer (2016) 446–462
2. Taheriyan, M., Knoblock, C.A., Szekely, P., Ambite, J.L.: Learning the semantics of structured data sources. Journal of Web Semantics **37** (2016) 152–169
3. Ramnandan, S.K., Mittal, A., Knoblock, C.A., Szekely, P.: Assigning semantic labels to data sources. In: European Semantic Web Conference, Springer (2015) 403–417
4. 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. In: Extended Semantic Web Conference, Springer (2012) 375–390
5. Cremaschi, M., De Paoli, F., Rula, A., Spahiu, B.: A fully automated approach to a complete semantic table interpretation. Future Generation Computer Systems (2020)
6. Zhang, Z.: Effective and efficient semantic table interpretation using tableminer+. Semantic Web **8**(6) (2017) 921–957

7. Kachroudi, M., Diallo, G., Ben Yahia, S.: OAEI 2017 results of KEPLER. In: Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017. Volume 2032 of CEUR Workshop Proceedings., CEUR-WS.org (2017) 138–145
8. Kachroudi, M., Ben Yahia, S.: Dealing with direct and indirect ontology alignment. J. Data Semant. **7**(4) (2018) 237–252
9. Kachroudi, M., Diallo, G., Ben Yahia, S.: KEPLER at OAEI 2018. In: Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. Volume 2288 of CEUR Workshop Proceedings., CEUR-WS.org (2018) 173–178
10. Ehrlinger, L., Wöß, W.: Towards a definition of knowledge graphs. SEMANTiCS (Posters, Demos, SuCCESS) **48** (2016) 1–4
11. Färber, M., Bartscherer, F., Menne, C., Rettinger, A.: Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. Semantic Web **9**(1) (2018) 77–129
12. Cremaschi, M., Avogadro, R., Chieregato, D.: Mantistable: an automatic approach for the semantic table interpretation. In: SemTab@ ISWC. (2019) 15–24
13. Ritze, D., Lehmberg, O., Bizer, C.: Matching html tables to dbpedia. In: Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics. (2015) 1–6