

# Yet Another Milestone for Kepler-aSI at SemTab 2022

Wiem Baazouzi<sup>1</sup>, Marouen Kachroudi<sup>2</sup> and Sami Faiz<sup>3</sup>

<sup>1</sup>Université de Manouba, Ecole Nationale des sciences de l'informatique, Laboratoire de Recherche en génie logiciel, Application distribuées, Manouba 2010, Tunis, Tunisie.

<sup>2</sup>Université de Tunis El Manar, Faculté des Sciences de Tunis, Informatique Programmation Algorithmique et Heuristique, LR11ES14, 2092, Tunis, Tunisie.

<sup>3</sup>Université de Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis, Laboratoire de Télédétection et Systèmes d'Information à Référence Spatiale, 99/UR/11-11, 2092, Tunis, Tunisie

## Abstract

In this paper, we present our system Kepler-aSI, for the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2022). This system is participating for the second time in this challenge edition, bringing improvements and new technical aspects. Kepler-aSI analyzes tabular data to be able to detect correct matches in Wikidata and Dbpedia. It should be noted that each data resource, or each round of the challenge imposes a certain number of constraints, requiring advanced techniques. The aforementioned task turns out to be difficult for the machines, which requires an additional effort in order to deploy the congenitive capacity in the matching methods. Kepler-aSI [1, 2, 3, 4] still relies on the SPARQL query to semantically annotate tables in Knowledge Graphs (KG), in order to solve the critical problems of matching tasks. The results obtained during the evaluation phase are encouraging and show the strengths of the proposed system.

## Keywords

Tabular Data, Knowledge Graph, Kepler-aSI, SPARQL

## 1. Introduction

It is evident that the World Wide Web encompasses and conveys very large volumes of textual information, in several forms: unstructured text, semi-structured model-based web pages (which represent data in the form widely recognized by key-value notation and lists), and of course arrays. In this broad context, the methods aiming to extract information from these resources to convert them in a structured form have been the subject of several works [5, 6]. As an observation, it is evident that there is a lack of understanding of the semantic structure which can hamper the process of data analysis. This observation reveals a gap between data islands. Indeed, acquiring this semantic reconciliation will therefore be very useful for data integration, data cleansing, data mining, machine learning and knowledge discovery tasks. For example, understanding the data can help assess the appropriate types of transformation.


Depending on the use and deployment scenario, tabular data is carefully conveyed to the Web in various formats. The majority of these datasets are available in tabular form (e.g., CSV

---

SemTab 2022

✉ [wiem.baazouzi@ensi-uma.tn](mailto:wiem.baazouzi@ensi-uma.tn) (W. Baazouzi); [marouen.kachroudi@fst.rnu.tn](mailto:marouen.kachroudi@fst.rnu.tn) (M. Kachroudi); [sami.faiz@insat.rnu.tn](mailto:sami.faiz@insat.rnu.tn) (S. Faiz)

ORCID  0000-0002-6512-7382 (W. Baazouzi); 0000-0002-7536-0428 (M. Kachroudi); 0000-0001-7065-6572 (S. Faiz)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

(Comma-Separated Values)). The main reason for the popularity of this format is its simplicity: many common office tools are available to facilitate their generation and use. Tables on the Web are a very valuable data source. Thus, injecting semantic information into arrays on the web has the potential to boost a wide range of applications, such as web searching, answering queries, and building Knowledge Bases (KB). Research reports that there are various issues with tabular data available on the Web, such as learning with limited labeled data, defining or updating ontologies, exploiting prior knowledge, and / or scaling up existing solutions. Therefore, this task is often difficult in practice, due to missing, incomplete or ambiguous metadata (*e.g.*, table and column names). 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) [7, 8, 9, 10, 11] or unsupervised (tables whose data is not dedicated to learning) [12, 11]. 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 [13, 14, 15, 16, 17].

This year's SemTab challenge differs from the last two sessions<sup>1 2</sup>, in that it deals with Wikidata and Dbpedia. In this challenge, the input is a CSV file, but three different challenges had to be met :

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

Data annotation is a fundamental process in tabular data analysis [18, 19], it allows to infer the meaning of other information. Then deduce the meaning of a tabular Knowledge Graph. The data we used was based both on Wikidata and Dbpedia. It should be noted that in a broader context, the data used and manipulated obey the triples format representation : subject ( $\mathcal{S}$ ), a predicate ( $\mathcal{P}$ ) and an object ( $\mathcal{O}$ ). This notation ensures semantic navigability through the data and makes all data manipulation more fluid, explicit and reliable. 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 either Wikidata or Dbpedia. Column Type Annotation (CTA) assigns connected semantic type to a column.

Our goal is to design a fast and efficient approach to annotate tabular data with entities from Wikidata or Dbpedia. Our approach combines a multitude of NLP and search and filter strategies, based on text preprocessing techniques. Experiments carried out in the context of SemTab 2022 for all tasks have shown encouraging results.

---

<sup>1</sup><https://www.cs.ox.ac.uk/isg/challenges/sem-tab/2019/>

<sup>2</sup><https://www.cs.ox.ac.uk/isg/challenges/sem-tab/2020/>

## 2. Kepler-aSI approach

In this section, we will describe in detail the different stages of our system, while presenting some basic notions to highlight the technical issues identified.

### 2.1. Key notions

- **Tabular Data** :  $S$  is a two-dimensional tabular structure made up of an ordered set of  $N$  rows and  $M$  columns, as depicted by Figure ?? .  $n_i$  is a row of the table ( $i = 1 \dots N$ ),  $m_j$  is a column of the table ( $j = 1 \dots M$ ). 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 ( $S$ ):  $M \times N$ .
- Subject Cell:  $S_{(i,0)}$  ( $i = 1, 2 \dots N$ ).
- Object Cell:  $S_{(i,j)}$  ( $i = 1, 2 \dots M$ ), ( $j = 1, 2 \dots N$ ).

$$\begin{array}{c}
 \text{Col}_0 \quad \quad \quad \text{Col}_i \quad \quad \quad \text{Col}_N \\
 \text{Row}_1 \quad \left( \begin{array}{ccccc} S_{1,0} & \dots & \dots & \dots & S_{1,N} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ S_{j,0} & \dots & S_{j,i} & \dots & S_{j,N} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ S_{M,0} & \dots & \dots & \dots & S_{M,N} \end{array} \right)
 \end{array}$$

Figure 1: Target Table

- **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 [20] 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 [20], authors agreed that a more precise definition was hard to find at that point. This statement points out the need of a 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) [21]. Although this definition is the only formal one, it contradicts with more general definitions as it explicitly requires the RDF data model. In the following we present a detailed description of our contribution, namely Kepler-aSI.

## 2.2. System description

In order to address the above mentioned SemTab challenge tasks, Kepler-aSI is designed according to the workflow depicted by Figure 2. There are three major complementary modules which consist in, respectively, Preprocessing, Annotation context and Tabular data to KG matching. The aforementioned steps are the same for each round, but the changes remain minimal depending on the variations observed in each case.

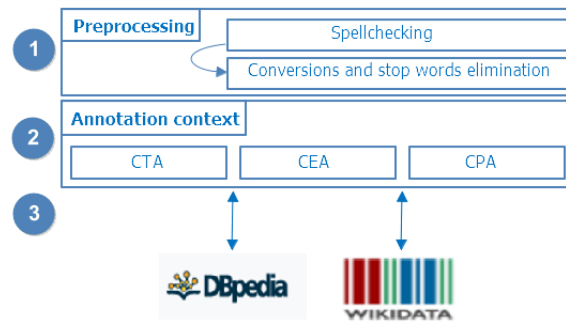


Figure 2: Kepler-aSI Workflow

As shown in Figure 2 Preprocessing aims to prepare the data inside the considered table. While Annotation Context, seeks to create a list of terms denoting the same context.

### 2.2.1. Preprocessing

It should be noted that the content of each table can be expressed according to different types and formats, namely: numeric, character strings, binary data, date/time, boolean, addresses, etc. Indeed, with the great diversity of data types, the pre-processing step is crucial. Therefore, the goal of preprocessing is to ensure that the processing of each table is triggered without errors. The effort is especially accentuated when the data contains spelling errors. In other words, these issues must be resolved before we apply our approach. In order to well carry out this step, we used several techniques and libraries such as (Textblob<sup>3</sup>, Pyspellchecker<sup>4</sup>, etc.) to rectify and correct all the noisy textual data in the considered tables. As an example, we detect punctuation, parentheses, hyphen and apostrophe, and also stop words by using the Pandas<sup>5</sup> library to remove them. Like a classic treatment in this register, we ended this phase by transforming all the upper case letters into lower case.

<sup>3</sup><https://textblob.readthedocs.io/en/dev/>

<sup>4</sup><https://pypi.org/project/pyspellchecker/>

<sup>5</sup><https://pandas.pydata.org>

## 2.2.2. Annotation context

This phase allows to explicitly extract the candidates for the annotation process. The priming is carried out by a processing columns analysis, which aims to understand and delimit the set of regular expressions which contains a set of units: the area, the currency, the density, the electric current, the energy, flow rate, force, frequency, energy efficiency, unit of information, length, density, mass, numbers, population density, power, pressure, speed, temperature, time, torque, voltage and volume. This step allows to identify multiple Regextypes using regular expressions (e.g. numbers, geographic coordinates, address, code, color, URL). Since all values of type text are selected, preprocessing for natural languages was performed using the langrid<sup>6</sup> library to detect 26 languages in our data. By the way, it's a novelty for this year's SemTab challenge, *i.e.*, which makes the task more difficult with the introduction of natural language barriers. The langrid library is a stand-alone language identification tool. It is performed on a large number of languages (97 currently). Doing so, correction, data type and language detection is performed. This can considerably reduce the effort and the cost of executing our approach by avoiding the massive repetition of these treatments for all the table cells, and this in each subtask.

## 2.2.3. Assigning a semantic type to a column (CTA)

As depicted by Figure 3, the task is to annotate each entity column with elements from Wikidata (or possibly Dbpedia) as its type identified during the preprocessing phase.

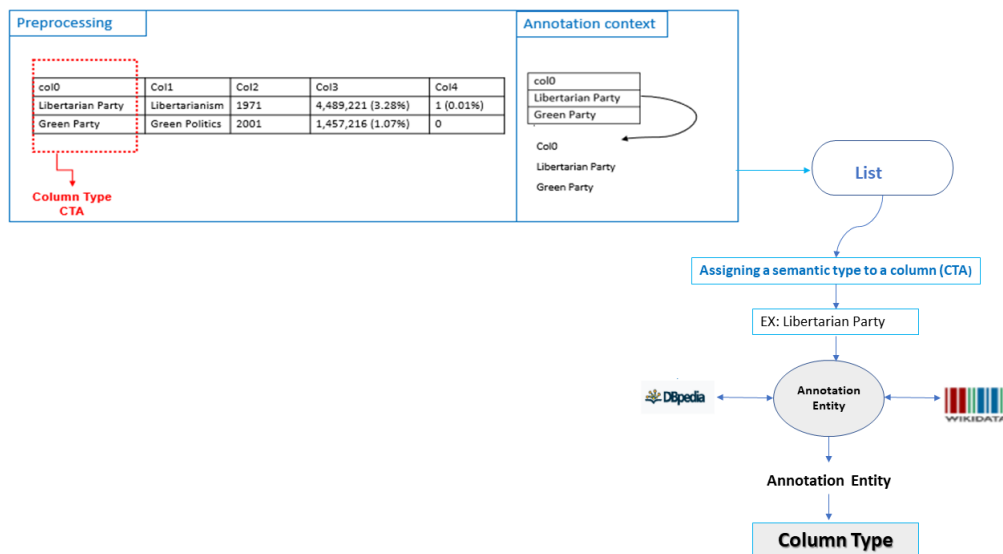


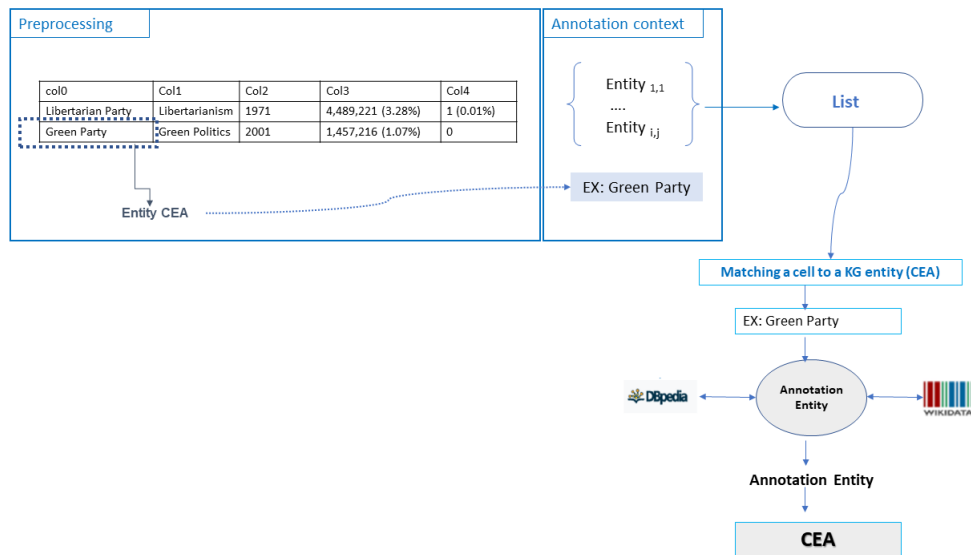
Figure 3: CTA task at a glance.

<sup>6</sup><https://github.com/openlangrid>

Each item is marked with the tag in Wikidata or Dbpedia. This treatment allows semantics identification. The CTA task can be performed based on Wikidata or Dbpedia APIs which allows us to search for an item according to its description. The main information collected about a given entity and used in our approach are: a list of instances (expressed by the `instanceOf` primitive and accessible by the P31 code), the subclass of (expressed by the `subclassOf` primitive and accessible by code P279) and overlaps (expressed by the `partOf` primitive and accessible by code P361). At this point, we are able to process the CTA task using a SPARQL query. The SPARQL query is our interrogation mean fed by the main information of the entity which governs the choice of each data type, since they are a list of instances (P31), of subclasses (P279) or a part of a class (P361). The result of the SPARQL query may return a single type but for some cases the result is more than one type, so in this case no annotation is produced for the CTA task.

#### 2.2.4. Matching a cell to a KG entity (CEA)

The CEA task aims to annotate the cells of a given table to a specific entity listed on Wikidata or Dbpedia.



**Figure 4:** Descriptive model of CEA task.

Figure 4 gathers the CEA task that can be performed based on the same principle of CTA task. Our approach reuses the results of the CTA task process by introducing the necessary modifications on the SPARQL query. If the operation returns more than one annotation and since we are conducting a fuzzy search [22, 23], we run a process based on examining the context of the considered column, relatively to what was obtained with the CTA task, to overcome the ambiguity problem.

### 2.2.5. Matching a property to a KG entity (CPA)

After having annotated the cell values as well as the different types of each of the considered entities, we will identify the relationships between two cells appearing on the same row via a property using a SPARQL query, as flagged by Figure 5. Indeed, the CPA task look for annotating the relationship between two cells in a row via a property. Similarly, this latter task can be performed in an analogous manner to the CTA and CEA tasks. The only difference in the CPA task is that the SPARQL query must select both the entity and the corresponding attributes. The properties are fairly easy to match since we have already determined them during CEA and CTA task processing.

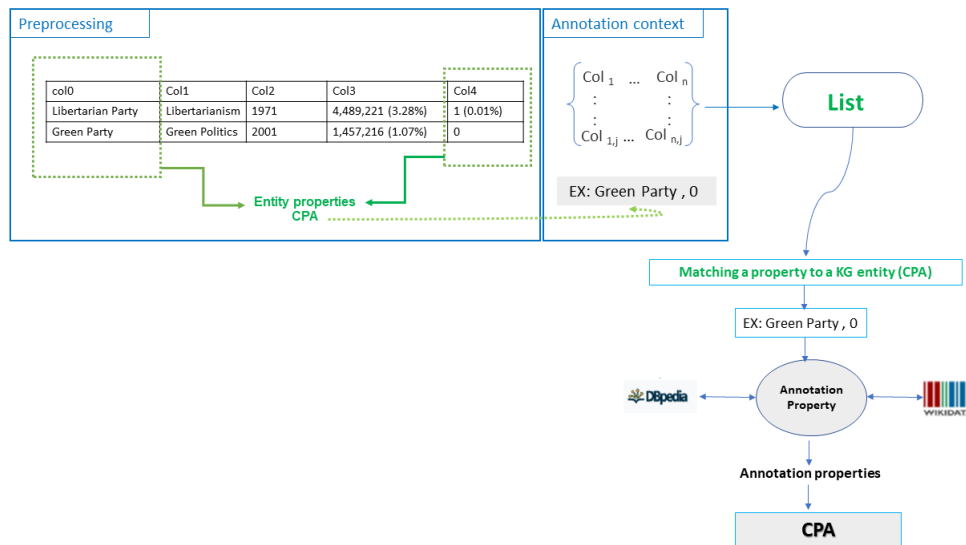


Figure 5: A representation of CPA task.

## 3. Kepler-aSI performance and results

In this section we will present the results of Kepler-aSI for the different matching tasks in the 3 rounds of SemTab 2022. These results highlight the strengths of Kepler-aSI with its encouraging performance despite the multiplicity of issues.

### 3.1. Round 1

In this first round, and for this version of SemTab 2022, three tasks are presented: CTA-WD, CEA-WD and CPA-WD. The column type annotation (CTA -WD) assigns a Wikidata semantic type (a Wikidata entity) to a column. Cell Entity Annotation (CEA-WD) maps a cell to a KG entity. Annotation should be represented by its full IRI, where case is not sensitive. Each line should include a column identified by a table ID and a column ID, along with the column annotation (a Wikidata item). This means that a row must include three fields: "Table ID", "Column ID", and "IRI Annotation", where:

- "*Table ID*" is the filename of the table data, but does not include the extension.
- "*Column ID*" is the position of the column in the input, starting from 0, *i.e.*, the ID of the first column is 0.
- "*IRI annotation*": the prefix of <http://www.wikidata.org/entity/> instead of <https://www.wikidata.org/wiki/> which is the URL prefix of the Wikidata page.

When it comes to associating a cell with an entity on the Knowledge Graph, the task is to annotate each target cell with an entity from Wikidata. Each annotation must contain the target cell annotation. A cell can be annotated by an entity with the prefix <http://www.wikidata.org/entity/>. Each CTA annotation must contain the annotation of a cell identified by a table identifier, a column identifier and a row identifier. Namely, an annotation must have four fields: "*Table ID*", "*Row ID*", "*Column ID*" and "*Entity IRI*", where:

- "*Table ID*": does not include the filename extension; and removing the .csv extension from the filename.
- "*Column ID*": is the position of the column in the table file, starting from 0, *i.e.* the ID of the first column is 0.
- "*Row ID*": is the row position in the table file, starting from 0, *i.e.* the first row ID is 0.
- "*Entity IRI*": the prefix of <http://www.wikidata.org/entity/> instead of <https://www.wikidata.org/wiki/> which is the URL prefix of the Wikidata page.

As for the annotation of column properties by Wikidata (CPA-WD), it consists in annotating the relations between the columns in a table with the properties of Wikidata. Each annotation must contain the two-column annotation which is itself identified by a table identifier, a first column identifier and a second column identifier. Namely, a row must have four fields: "*Table ID*", "*Column ID 1*", "*Column ID 2*" and "*Property IRI*". Each pair of columns must be annotated by at most one property, as follows:

- "*Table ID*" does not include filename extension
- "*Column ID 1*", "*Column ID 2*", is the position of the column in the table file, starting from 0, *i.e.* the first column ID is 0.
- "*Property IRI*": prefix of <http://www.wikidata.org/prop/direct/> instead of <https://www.wikidata.org/wiki/> which is the URL prefix of the Wikidata page.

It should be noted that the CTA-WD, CEA-WD and CPA-WD task data contains 3691 tables. Results are summarized in Table 1:

### 3.2. Round 2

Round 2 includes 3 main families of tests, the results of which are summarized in Table 2:

- HardTables (HT-WD): represented by 4649 tables;
- ToughTablesR2-WD (2T-WD): represented by 114 tables;
- ToughTablesR2-DBP (2T-DBP): represented by 114 tables.



<i>Round 1</i>								
CTA			CEA			CPA		
APrecision	AF1	Rank	APrecision	AF1	Rank	APrecision	AF1	Rank
0.944	0.944	3/10	—			0.937	0.937	4/10

**Table 1**  
Results for Round 1

<i>Round 2</i>			
	APrecision	AF1	Rank
<b>HardTables-CTA-WD</b>	0.881	0.811	3/6
<b>HardTables-CEA-WD</b>	—		
<b>HardTables-CPA-WD</b>	0.912	0.912	3/6
<b>ToughTables-CTA-WD</b>	0.369	0.369	3/6
<b>ToughTables-CEA-WD</b>	—		
<b>ToughTables-CTA-DBP</b>	0.154	0.154	5
<b>ToughTables-CEA-DBP</b>	—		

**Table 2**  
Results for Round 2

### 3.3. Round 3

Round 3 includes 3 main families of tests, metrics are in Table 3:

- GitTables schema: represented by 45 tables;
- GitTables DBP: represented by 6898 tables;
- Bio-Div-Tables: represented by 45 tables.

<i>Round 3</i>			
	APrecision	AF1	Rank
<b>BiodivTab-CTA-DBP</b>	0.781	0.731	3/7
<b>BiodivTab-CEA-DBP</b>	0.534	0.534	4/7
<b>GitTables-CTA-DBP</b>	—		
<b>GitTables-CTA-SCH</b>	—		

**Table 3**  
Results for Round 3

In Round 3, we realized that there were significant amounts of entity duplication in our result. Thus, the pairing process has been improved by adding the following features. First,

spell checking of misspelled sentences was used. However, approaches based on relishing content duplications can achieve results without column duplication. In order to overcome duplicate columns, We used Fuzzy matching in pandas to detect duplicate rows (efficiently). In fact, FuzzyWuzzy is an implementation of edit distance, which would be a good candidate for constructing a pairwise distance matrix in numpy or similar. To detect "duplicates" or close matches, We have to compare each row to the other rows or We will never know if two are close to each other

## 4. Conclusion

To conclude, we have presented in this paper the second version of our Kepler-aSI approach. Our system is participating in the challenge for the third time, it is approaching maturity and achieving very encouraging performance. We have succeeded in combining several strategies and treatment techniques, which is also the strength of our system. We boosted the preprocessing and spellchecking steps that got the system up and running.

In addition, despite the data size, which is quite large, we managed to get around this problem by using a kind of local dictionary, which allows us to reuse already existing matches. Thus, we realized a considerable saving of time, which allowed us to adjust and rectify after each execution. We also participated in all the tasks without exception, which allowed us to test our system on all facets, *i.e.*, to identify its strengths and weaknesses.

In this paper, we presented our contribution to the SemTab2022 challenge, Kepler-aSI. We tackled the several proposed tasks. Our solution is based on a generic SPARQL query using the cell contents as a description of a given item. In each round, despite the time allocated by the organizers running out, we continued the work and the improvements, having the conviction that each effort counts and brings us closer to the good control of the studied field.

## References

- [1] W. Baazouzi, M. Kachroudi, S. Faiz, Kepler-asi: Kepler as a semantic interpreter, in: Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2020) co-located with the 19<sup>th</sup> International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 5, 2020, volume 2775, 2020, pp. 50–58.
- [2] W. Baazouzi, M. Kachroudi, S. Faiz, Kepler-asi at semtab 2021, in: Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching co-located with the 20<sup>th</sup> International Semantic Web Conference (ISWC 2021), Virtual conference, October 27, 2021, volume 3103, 2021, pp. 54–67.
- [3] W. Baazouzi, M. Kachroudi, S. Faiz, Towards an efficient fairification approach of tabular data with knowledge graph models, in: Proceedings of the 26<sup>th</sup> Knowledge-Based and

- Intelligent Information Engineering Systems International Conference KES 2022, volume 207, 2022, pp. 2727–2736.
- [4] W. Baazouzi, M. Kachroudi, S. Faiz, A matching approach to confer semantics over tabular data based on knowledge graphs, in: Proceedings of the 11<sup>th</sup> International Conference on Model and Data Engineering, Springer, 2023, pp. 236–249.
  - [5] J. Chen, E. Jiménez-Ruiz, I. Horrocks, C. Sutton, Colnet: Embedding the semantics of web tables for column type prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 29–36.
  - [6] S. Malyshev, M. Krötzsch, L. González, J. Gonsior, A. Bielefeldt, Getting the most out of wikidata: Semantic technology usage in wikipedia’s knowledge graph, in: International Semantic Web Conference, Springer, 2018, pp. 376–394.
  - [7] M. Pham, S. Alse, C. A. Knoblock, P. Szekely, Semantic labeling: a domain-independent approach, in: International Semantic Web Conference, Springer, 2016, pp. 446–462.
  - [8] M. Taheriyani, C. A. Knoblock, P. Szekely, J. L. Ambite, Learning the semantics of structured data sources, *Journal of Web Semantics* 37 (2016) 152–169.
  - [9] S. K. Ramnandan, A. Mittal, C. A. Knoblock, P. Szekely, Assigning semantic labels to data sources, in: European Semantic Web Conference, Springer, 2015, pp. 403–417.
  - [10] C. A. Knoblock, P. Szekely, J. L. Ambite, A. Goel, S. Gupta, K. Lerman, M. Muslea, M. Taheriyani, P. Mallick, Semi-automatically mapping structured sources into the semantic web, in: Extended Semantic Web Conference, Springer, 2012, pp. 375–390.
  - [11] M. Cremaschi, F. De Paoli, A. Rula, B. Spahiu, A fully automated approach to a complete semantic table interpretation, *Future Generation Computer Systems* (2020).
  - [12] Z. Zhang, Effective and efficient semantic table interpretation using tableminer+, *Semantic Web* 8 (2017) 921–957.
  - [13] M. Kachroudi, G. Diallo, S. Ben Yahia, 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, pp. 138–145.
  - [14] M. Kachroudi, S. Ben Yahia, Dealing with direct and indirect ontology alignment, *J. Data Semant.* 7 (2018) 237–252.
  - [15] M. Kachroudi, G. Diallo, S. Ben Yahia, 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, pp. 173–178.
  - [16] M. Kachroudi, S. Zghal, S. Ben Yahia, Bridging the multilingualism gap in ontology alignment, *International Journal of Metadata, Semantics and Ontologies* 9 (2014) 252–262.
  - [17] M. Kachroudi, S. Zghal, S. Ben Yahia, Using linguistic resource for cross-lingual ontology alignment, *International Journal of Recent Contributions from Engineering* 1 (2013) 21–27.
  - [18] J. Chen, E. Jiménez-Ruiz, I. Horrocks, C. Sutton, Learning semantic annotations for tabular data, arXiv preprint arXiv:1906.00781 (2019).
  - [19] V. Efthymiou, O. Hassanzadeh, M. Rodriguez-Muro, V. Christophides, Matching web tables with knowledge base entities: from entity lookups to entity embeddings, in: International Semantic Web Conference, Springer, 2017, pp. 260–277.
  - [20] L. Ehrlinger, W. Wöß, Towards a definition of knowledge graphs., SEMANTiCS (Posters,

Demos, SuCCESS) 48 (2016) 1–4.

- [21] M. Färber, F. Bartscherer, C. Menne, A. Rettinger, Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago, *Semantic Web* 9 (2018) 77–129.
- [22] H. Akremi, S. Zghal, Dof: a generic approach of domain ontology fuzzification, *Frontiers Comput. Sci.* 15 (2021) 153322.
- [23] H. Akremi, M. G. Ayadi, S. Zghal, To medical ontology fuzzification purpose: Covid-19 study case, *Procedia Computer Science* 207 (2022) 1027–1036.