Kepler-aSI : Semantic Annotation for Tabular Data

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

[1] [a]RIADI Laboratory, National School of Computer Science, University of Manouba,
Manouba 2010, Tunisia
[b]ESPRIT School of Engineering, El Ghazala 2080, Tunisia.
wiem.baazouzi@ensi-uma.tn
[2] University of Tunis El Manar, Faculty of Sciences of Tunis, Computer Science,
Algorithmic Programming and Heuristics, LR11ES14, 2092, Tunis, Tunisia
marouen.kachroudi@fst.rnu.tn
[3] University of Tunis El Manar, National School of Engineers of Tunis,
Laboratory of Remote Sensing and Spatial Reference Information Systems,
99/UR/11-11, 2092, Tunis, Tunisia
sami.faiz@insat.rnu.tn

Abstract. The identification of semantic concepts in tabular data is crucial for numerous applications, including data integration, cleaning, retrieval, feature engineering, and model development in machine learning. Recently, various studies have introduced methods using supervised learning or heuristic models to annotate semantic types. However, these approaches have limitations, making it difficult for them to generalize to a wide range of concepts or examples. Additionally, many neural network-based methods struggle with scalability, and the majority of the existing techniques do not perform well with numerical data. We present Kepler-aSI, a column-to-concept mapping technique that employs a maximum likelihood estimation approach through sets. This method effectively leverages large amounts of publicly available table data, even in the presence of some noise. We showcase the effectiveness of Kepler-aSI in the Semtab2024 challenge.

Keywords: Tabular Data - Knowledge Graph - Kepler-aSI - SPARQL

1 Introduction

Semantic annotation of structured data plays a vital role in various applications, from information retrieval and data preparation to training classifiers. For instance, schema matching in data integration demands accurate identification of column types in input tables [38]. Similarly, automated data cleaning and transformation techniques use semantic types to establish validation rules [30]. Tasks like dataset discovery [46] and feature acquisition in machine learning [24] depend on assessing the semantic similarity of entities across multiple tables. Many commercial tools, including Google Data Studio [34], Microsoft Power BI [37], and Tableau [12], leverage these annotations to interpret input data, detect inconsistencies, and create visualizations. Semantic annotation of a table column involves identifying real-world concepts that represent the data's meaning. While this process is critical for numerous data science applications, most systems currently rely on regular expression or rule-based techniques to identify column types. These methods necessitate predefined models, struggle with noisy datasets, and fail to generalize beyond the input models. Recently, there has been increasing interest in applying deep learning techniques to detect semantic types, due to their robustness against noisy data and superiority over traditional rule-based systems. Earlier approaches can be divided into two categories based on the training data used and the types of concepts they identify.

(a) *Knowledge Graphs*: ColNet [15] and HNN [16] are among the latest methods that utilize semantic types derived from knowledge graphs like Wikidata. These techniques generate candidate types and train classifiers to estimate the likelihood of each candidate type. However, they mainly identify semantic types for columns partially present in the knowledge graphs and struggle to generalize to broader categories, such as person names.

(b) *Tabular Data*: Sherlock [26] and Sato [45] are recent approaches that treat the task of labeling concepts as a multi-class classification problem. These methods train classifiers using open data but are confined to concepts that precisely match the predefined list of Wikidata concepts.

To address the limitations of prior work, we make the following observations :

1. There is a plethora of publicly available structured data from diverse sources such as data.gov, Wikipedia tables, explored web table collections, and others, as well as knowledge graphs like DBPedia and Wikidata;
2. It is true that not all sources are well-organized, and some degree of noise within each source must be considered. However, a robust ensemble from multiple input sources can help eliminate noise. While a strict classification modeling method may require reference data, a carefully designed probability estimation method can be more tolerant of noise and scale with larger data contents;
3. Numerical data requires special treatment compared to categorical entity data. Although a numerical value is less unique than a named entity (e.g., 20 can mean several things), a group of numbers representing a certain concept follows particular patterns. The use of meta-features such as range and distribution aids in the rapid identification of numerical concepts and is robust to small amounts of noise;
4. Instead of considering each column of the table in isolation, the overall context of the dataset, combined with information from a knowledge graph (KG), allows for the joint estimation of the probability of correspondence between the KG concepts and the attributes of the tabular data. This improves the identification of links and similarities by taking into account the relationships between the concepts in the KG and the structures of the tabular data.

The Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab) was established to evaluate systems by proposing different tasks and datasets [2, 3, 19, 27, 28, 25]. Systems use various methods to generate annotations, either by analyzing large knowledge bases [2, 3, 4, 19, 27, 28, 25] or by using classification based on training examples [22, 40].

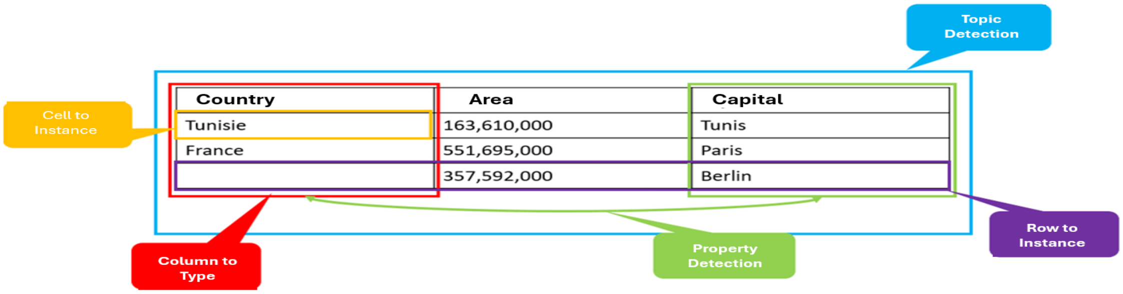
We develop our solution within the Kepler-aSI system[9, 8, 6], and we present the results obtained from its use in the SemTab challenge on tabular data to knowledge graph matching, as described in Section 2 & 3. This challenge evaluates various semantic annotation methods on large-scale tabular data. In Section 4, we describe our approach by detailing the similarities and differences with other competing approaches.

2 SemTab Challenge

The Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab) evaluates table annotation systems on various datasets and annotation tasks [20]. In its sixth edition (SemTab 2024), it consists of two rounds, each featuring a variety of tables to be annotated with concepts from Wikidata. The evaluation of system accuracy follows a similar approach to previous versions of SemTab. Specifically, SemTab2024[4] is based on using typical multi-class classification metrics, as detailed below. Additionally, for the CTA task, we adopt the ”cscore” metric to reflect the distance in the type hierarchy between the predicted column type and the ground truth semantic type.

The challenge is divided into five annotation tasks (Figure 1):

[4] https://sem-tab-challenge.github.io/2024/tracks/accuracy-track.html



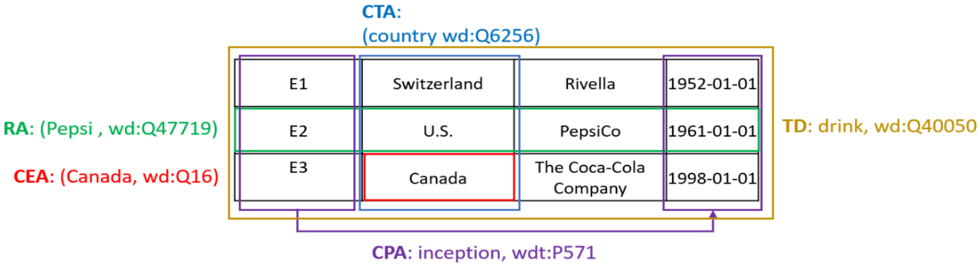
Country	Area	Capital
Tunisie	163,610,000	Tunis
France	551,695,000	Paris
	357,592,000	Berlin

Fig. 1. SemTab tasks

- **CTA Task**: Assigning a semantic type (a Wikidata class as fine-grained as possible) to a column.
- **CEA Task**: Matching a cell to a Wikidata entity.
- **CPA Task**: Assigning a Wikidata property to the relationship between two columns.
- **RA Task**: Assigning a Wikidata entity to a table row.
- **TD Task**: Assigning a Wikidata class to a table.

2.1 Table Types :

- **Horizontal Tables :** A grid where each row represents one entity and each column shares the same semantic type .



E1	Switzerland	Rivella	1952-01-01
E2	U.S.	PepsiCo	1961-01-01
E3	Canada	The Coca-Cola Company	1998-01-01

Fig. 2. Horizontal Table Example

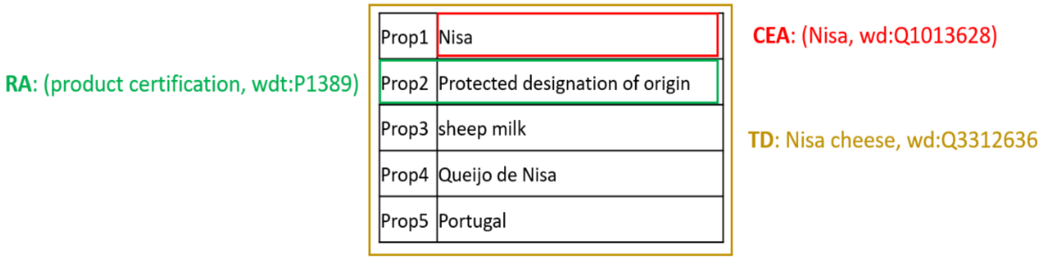
- **Entity Tables :** A list where rows describe different properties of a single entity, with each row representing a property of that entity.

3 Related Work

In this section, we discuss prior technqiues that have been developed to identify the type of a column.

3.1 Regular Expression and Lookup-Based Techniques

Many techniques have been developed to identify the type of a column in semantic tables, often relying on regular expressions and lookup-based methods. These techniques use manually defined patterns to



Prop1	Nisa
Prop2	Protected designation of origin
Prop3	sheep milk
Prop4	Queijo de Nisa
Prop5	Portugal

CEA: (Nisa, wd:Q1013628)
RA: (product certification, wdt:P1389)
TD: Nisa cheese, wd:Q3312636

Fig. 3. Entity Table Example

identify column types, which play a critical role in various data science pipelines such as feature enrichment [24], schema mapping, data cleaning and transformation [29, 30], and structured data search [23]. However, the manual effort required for enumerating these patterns can be significant. To reduce this manual effort, some techniques perform fuzzy lookups of each cell value over knowledge graphs to identify concepts [18, 24, 28]. These methods assume that the cell values are present in the knowledge graphs and are not robust to noise. Deng et al. [21] presented a scalable method based on fuzzy matching between entities for a concept and the cell values of a column, using similarity scores for ranking. However, this approach can lack robustness, as it might confuse similar entities (e.g., movies and novels with the same name). For numerical columns, Neumaier et al. [35] developed a method that clusters values and uses nearest neighbor search to identify the most likely concept. This method does not leverage column metadata and the context of co-occurring columns.

3.2 Graphical Models

Advanced concept identification techniques generate features for each input column and use probabilistic graphical models to predict labels. Limaye et al. [31] use a graphical model to collectively determine cell annotations, column annotations, and binary column relationships. While effective, these techniques can be sensitive to noisy values and might not capture semantically similar values, which have been successfully captured by recent word embedding-based techniques.

3.3 Learning Approaches Using Neural Networks

Chen et al. [15] introduced ColNet, a CNN-based approach for classification. It constructs positive and negative examples by looking up cell values over DBPedia, embedding these examples using word2vec to train the CNN. This approach helps build context among different cells in the column. Chen et al. [16] extended this method to leverage inter-column semantics using a hybrid neural network (HNN), though this technique is slow, requiring extensive training time. Sherlock [26] models concept identification as a multi-class classification problem, training a multi-input feed-forward deep neural network over a large corpus of open data containing more than 600K columns referring to 78 semantic types. However, its reliance on large amounts of training data limits its applicability to less common concepts. SATO [45] builds on Sherlock by using context from co-occurring columns to jointly predict the concept of all columns in a dataset, treating the table as a document to generate a vector of terms representative of the table context. Nevertheless, its effectiveness is limited when column ordering is irrelevant.

3.4 Semantic Table Interpretation Systems

This section examines the literature by analyzing various contributions, focusing on the tasks of Column Type Annotation (CTA), Column Entity Annotation (CEA), and Column Pair Annotation (CPA). The methods are discussed from different perspectives: strengths, gaps, and the impact of table elements and/or the knowledge graph ($\mathcal{KG}$) structure on performance metrics. Various research works have tackled the issue of Semantic Table Interpretation (STI), varying in their deployed techniques and adopted approaches.

TabEL [13] begins with preprocessing, generates candidates for each cell using the YAGO ontology, and ranks them according to their string similarity with the cell. An undirected probabilistic graph model is then generated to capture the contextual co-occurrences of the entities. ADOG [36] uses an aggregation of string-based similarity, the number of property occurrences, and the normalized score of the Elasticsearch tool for each match via DBPedia. Tabularisi [41] relies on a statistical approach using TF-IDF to rank candidates for the CEA task. The scores are aggregated from TF-IDF, Levenshtein similarity, and word similarity. Magic [39] uses comparison matrices called INK embeddings to improve computational efficiency and perform CEA, CPA, and CTA annotations. It also introduces the concept of a key column for annotation. LOD4ALL [33] uses an RDF storage database and a score database to generate candidates and performs CTA, CEA, and CPA tasks after filtering the CTA results. CSV2KG [43] goes through six phases, including raw cell annotations, candidate disambiguation, and inferring column types and properties between columns. LinkingPark [17] uses a cascading approach to generate candidate entities and property links for annotation. DAGOBAH [14] consists of sequential tools to identify semantic relationships, enrich knowledge graphs, and produce metadata for reference. JenTab [1] operates through nine modules to generate and filter candidates for CTA, CEA, and CPA tasks using various filtering and solution selection strategies. LexMa [42] starts with preprocessing, evaluates lexical matching based on cosine similarity, and uses Wikidata and DBPedia search services. MantisTable [18] categorizes columns, generates candidates from SPARQL queries, performs cross-compatibility analysis for CEA, and uses majority voting for CPA. DAGOBAH Embeddings [14] proposes an annotation vision based on embedding vector spaces, using K-means clustering and *TransE* embeddings. Radar Station [32] uses graph embedding to detect latent relationships between entities and improve disambiguation. TCN [44] exploits intra and inter-table contextual information for CTA and CPA tasks, using transfer learning and unsupervised BERT-like pre-training. DODUO [40] learns to annotate relationships between column type and column pair by injecting table contexts into the prediction process, using column representations and token sequences.

4 KEPLER-aSI approach

In this section, we will provide a detailed description of our system and highlight some fundamental concepts related to the technical challenges we have identified. To tackle the tasks presented in the SemTab challenge, our system, KEPLER-aSI [10, 11, 7], follows the workflow illustrated in Figure 4. It consists of five main nested modules, namely Preprocessing, Query Engine, (and/or External Resource Consultation), $\mathcal{KG}$ Candidate Filtering, Annotation, and File Generation. While the overall steps remain the same for each round, minor adjustments may be made based on specific variations observed in each case.

4.1 Module 1: Pre-processing

1. **Data Sources :** Data is extracted from two main sources:

   – *Data Lakes :* Tables containing various columns, for example, a country table with columns such as 'Country Name,' 'Population,' and 'Capital.'
   – *Knowledge Graphs :* Databases like DBPedia and Wikidata, providing structured information about entities such as countries and their attributes.

2. **Data Extraction and Categorization** Columns are processed according to their specific types:

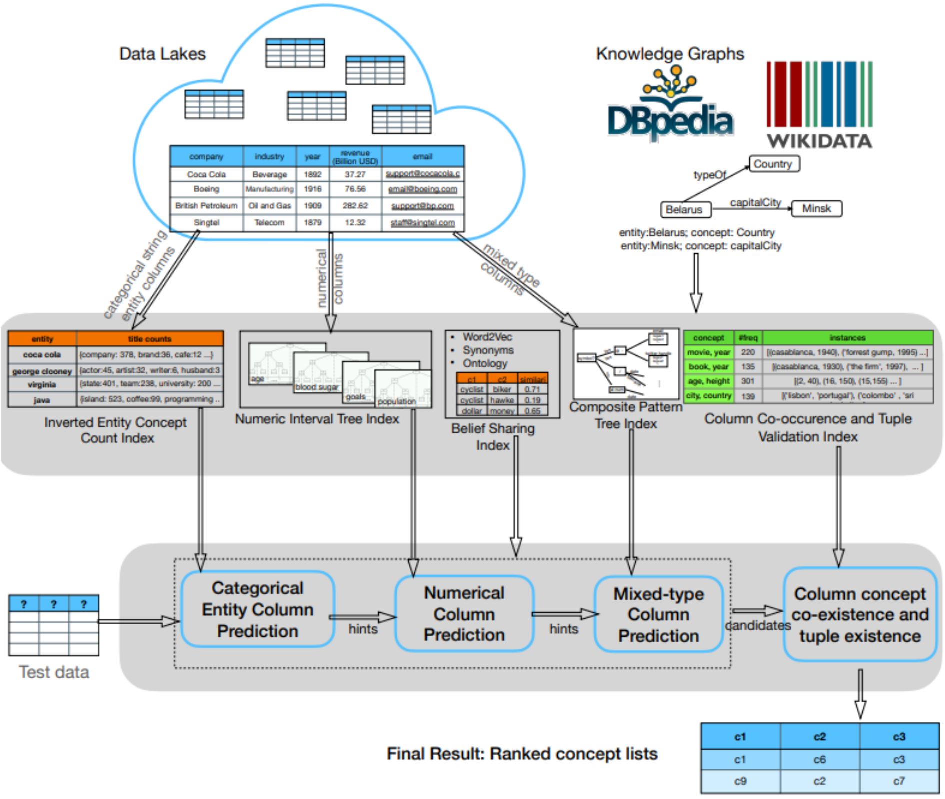

Data Lakes
company | industry | year | revenue (Billion USD) | email
Coca Cola | Beverage | 1892 | 37.27 | support@cocacola.c
Boeing | Manufacturing | 1916 | 76.56 | email@boeing.com
British Petroleum | Oil and Gas | 1909 | 282.62 | support@bp.com
Singtel | Telecom | 1879 | 12.32 | staff@singtel.com
Knowledge Graphs
DBpedia
WIKIDATA
Country
typeOf
Belarus
capitalCity
Minsk
entity:Belarus; concept: Country
entity:Minsk; concept: capitalCity
categorical string entity columns
numerical columns
mixed type columns
entity | title counts
coca cola | {company: 378, brand:36, cafe:12 ...}
george clooney | {actor:45, artist:32, writer:6, husband:3
virginia | {state:401, team:238, university: 200 ...
java | {island: 523, coffee:99, programming ..
Inverted Entity Concept Count Index
age
blood sugar
goals
population
Numeric Interval Tree Index
Word2Vec
Synonyms
Ontology
c1 | c2 | similari
cyclist | biker | 0.71
cyclist | hawke | 0.19
dollar | money | 0.65
Belief Sharing Index
Composite Pattern Tree Index
concept | #freq | instances
movie, year | 220 | [(casablanca, 1940), ('forrest gump, 1995) ...]
book, year | 135 | [(casablanca, 1930), ('the firm', 1997), ...]
age, height | 301 | [(2, 40), (16, 150), (15,155) ...]
city, country | 139 | [('lisbon', 'portugal'), ('colombo' , 'sri
Column Co-occurence and Tuple Validation Index
? ? ?
Test data
Categorical Entity Column Prediction
hints
Numerical Column Prediction
hints
Mixed-type Column Prediction
candidates
Column concept co-existence and tuple existence
Final Result: Ranked concept lists
c1 | c2 | c3
c1 | c6 | c3
c9 | c2 | c7

Fig. 4. Kepler-aSI Approach

- *Categorical String Columns :*
  - For example, for a 'Country Name' column, extracting entities like 'France' and 'Canada.'
  - Creating a concept index by counting the occurrences of each country.

- *Numerical Columns :*
  - For a 'Population' column, structuring values into intervals (e.g., 1 to 10 million, 10 to 50 million).

- *Mixed-Type Columns :*
  - For example, a column containing combinations of country names and capitals (such as 'France - Paris'), where techniques for both categorical and numerical columns are combined.

3. **Indexing :** Various indices facilitate prediction and annotation:

- *Inverted Entity Concept Count Index:*
  - Associates each country (e.g., 'France') with related concepts, such as population and capital, along with their occurrences in the data.

- *Numeric Interval Tree Index:*
  - Structures population data into an interval tree, for example, '1-10 million' and '10-50 million,' for efficient analysis.

- *Belief Sharing Index:*
    - Uses tools like Word2Vec to create similarity vectors, for instance, to understand that 'France' and 'Germany' are geographical entities.
- *Composite Pattern Tree Index:*
    - Generates models linking data columns, such as the association between 'Country Name' and 'Population.'
- *Column Co-occurrence and Tuple Validation Index:*
    - Validates relationships between concepts using knowledge graphs, such as 'France - capitalCity - Paris.'

4. **Prediction Models :** Specific models predict column types and annotate cells:
    - *Categorical Entity Column Prediction:*
        - For a 'Country Name' column, using the entity concept index to predict country names in the test data.
    - *Numerical Column Prediction:*
        - For a 'Population' column, using the numeric interval tree indices to predict value ranges in the test columns.
    - *Mixed-Type Column Prediction:*
        - For a mixed-type column, using belief sharing indices and composite models to predict data combinations.
5. **Validation and Ranking :**
    - *Column Concept Co-existence and Tuple Validation:*
        - Validate the predicted concepts and rank annotations using the column co-occurrence and tuple validation index.
    - *Final Result:*
        - Generate ranked lists of concepts for each column in the test data, providing precise annotations for each column type.

4.2 Module 2: Semantic Annotation

After performing the various pre-processing treatments, the tabular data annotation phase can be triggered.

Query Engine Sub-module The annotation module acts as the core component of the annotation phase. It allows us to extract candidate annotations from Knowledge Graphs (such as Wikidata) using parameterized SPARQL queries. At the beginning of the annotation phase, a switcher component examines the annotation context and determines the appropriate query to execute. This process is further detailed in the following section.

Example 1. Starting from an English entity description, below is an example of a SPARQL query to retrieve the label, class name, and properties from Wikidata (or possibly DBPedia):

```

    endpoint_url = "https://query.wikidata.org/sparql"
    query = """
   SELECT ?itemLabel ?class  ?property
   WHERE {
       ?item  ?itemDescription "%s"@en .
       ?item wdt:P31 ?class
     }"""
```

Assigning a Semantic Type to a Column (CTA) The task is to annotate each entity column with elements from Wikidata (or possibly DBPedia) as its type, identified during the preprocessing phase.

Algorithm 1: CTA task

```
Input: Table $\mathcal{T}$
Output: Annotated Table $\mathcal{T}'$
$i \leftarrow 0$
while $col_i \in \mathcal{T}$ do
    $class\_annot \leftarrow \emptyset$ /* Assigning a semantic type to a column. */
    while $cell \in col$ do
        $Label \leftarrow cell.expressionValue$
        $CorrectedLabel \leftarrow SpellCheckEngine(Label)$
        $\mathcal{KG}\_candidates \leftarrow QueryEngine(CorrectedLabel)$
        $class\_annot \leftarrow \mathcal{KG}\_candidates$
    Annotate($\mathcal{T}'.col_i, getBestRankedClass(class\_annot)$)
```

To annotate each entity column with elements from Wikidata, we utilize the tags associated with each item in Wikidata. This approach enables the identification of semantic information. The `CTA` task is accomplished by using the Wikidata APIs to search for an item based on its description. During the pre-processing phase, we collect essential information about each entity from Wikidata, including its instance list (indicated by the `instanceOf` primitive and identified by the P31 code), its subclasses (indicated by the `subclassOf` primitive and identified by the P279 code), and its overlaps with other classes (indicated by the `partOf` primitive and identified by the P361 code). To perform the `CTA` task, we use a SPARQL query that leverages this entity information. The SPARQL query serves as an interrogation tool, utilizing the gathered information about the entity's instances (P31), subclasses (P279), or class overlaps (P361) to determine the appropriate data type. The result of the SPARQL query may yield a single type. However, in cases where multiple types are returned, a disambiguation treatment is carried out to resolve the ambiguity. The syntax for the SPARQL query is as follows:

```

PREFIX rdfs: <http://wikidata.org/resource/>
SELECT ?item ?itemLabel ?class
WHERE {
        ?item ?itemDescription "%s"@en .
        ?item   wdt:P31 ?class
      }
```

Code Listing 1.1. The SPARQL query for the CTA task.

To ensure efficient and fast information retrieval, all the candidates obtained from the query are indexed using effective techniques. Each identified annotation is indexed and stored in a NoSQL database, specifically MongoDB[5]. This allows for efficient storage and retrieval of the annotations. The final annotation, which represents the result of the matching process, is determined by querying this MongoDB database through its integrated search engine. MongoDB was chosen as the database solution due to its ability to handle nested structures, which is important for organizing the annotations. Additionally, MongoDB offers significant performance benefits, such as scalability and efficient search capabilities, resulting in improved execution times. By using MongoDB, we can leverage its processing capabilities and benefit from its efficient scaling and search efficiency, ensuring the smooth and effective retrieval of annotations during the matching process.

[5] https://www.mongodb.com/docs/

Assigning a Cell to a $\mathcal{KG}$ Entity (CEA) The `CEA` task focuses on annotating the cells of a given table with specific entities listed on Wikidata or DBPedia. This task follows the same principle as the `CTA` task. Algorithm **??** provides an overview of the `CEA` task.

Algorithm 2: CEA task

```
Data: Table $\mathcal{T}$
Result: Annotated Table $\mathcal{T}'$
$i \leftarrow 0$
while $row_i \in \mathcal{T}$ do
    $entity\_annot \leftarrow \emptyset$ /* Matching a cell to a Wikidata entity. */
    while $cell \in row$ do
        $Label \leftarrow cell.expressionValue$
        $CorrectedLabel \leftarrow SpellCheckEngine(Label)$
        $\mathcal{KG}\_candidates \leftarrow QueryEngine(CorrectedLabel)$
        $entity\_annot \leftarrow \mathcal{KG}\_candidates$;
    Annotate($\mathcal{T}'$.row$_i$, $getBestRankedEntity(entity\_annot)$)
```

Our approach reuses the results of the `CTA` task process by introducing the necessary modifications to the SPARQL query. If the operation returns more than one annotation, we run a treatment based on examining the context of the considered column, relative to what was obtained with the `CTA` task, to overcome the ambiguity problem. Analogously to the CTA task, the CEA task can be performed through a SPARQL query as in the listing. The CEA task aims to annotate the cells of a given table to a specific entity listed on Wikidata or eventually another $\mathcal{KG}$. Moreover, if the concerned cells belong to columns already annotated during the CTA task, then their result can be reused by making the necessary adjustments. The process is the same as the CTA task. If the return value of the SPARQL query is a single candidate, then this candidate is retained as an annotation. If we get more than one candidate, the list is further processed to disambiguate. The SPARQL query syntax is as follows:

```

PREFIX rdfs: <http://wikidata.org/resource/>
SELECT ?object ?objectLabel ?class
WHERE {
        ?object ?objectDescription "%s"@en .
        ?object   wdt: P31 ?class
        }
```

Code Listing 1.2. The SPARQL query dealing with the CEA task.

Matching a Property to a $\mathcal{KG}$ Entity (CPA) After annotating the cell values as well as the different types of each of the considered entities, we identify the relationships between two cells appearing on the same row via a property using a SPARQL query, as detailed by Algorithm 3. Indeed, the CPA task involves annotating the relationship between two cells in a given row via a property. Similarly, this task is performed analogously 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 as depicted by the following listing:

```

PREFIX rdfs: <http://wikidata.org/resource/>
SELECT ?item1 ?property ?item2
WHERE {
        BIND(wdt:P279 AS ?property)
        ?item1 ?property ?item2.
        OPTIONAL { ?item1 wdt:P31 ?class. }
        OPTIONAL { ?item2 wdt:P31 ?class. }
      }
```

Code Listing 1.3. The SPARQL query designed for the CEA task.

The properties are easy to match since we have already detected them during CEA and CTA task processing.

Algorithm 3: CPA task

```
Data: Table $\mathcal{T}$
Result: Annotated Table $\mathcal{T}'$
$i \leftarrow 0$ $j \leftarrow 0$
while $(col_i, col_j) \in \mathcal{T} andi \neq j$ do
    $property\_annot \leftarrow \emptyset$ /* Assigning a KG property to the relationship between two columns. */
    $\mathcal{KG}\_class\_Label_1 \leftarrow Annotate(\text{T'}.col_i, getMostFrequentClass(class\_annot))$
    $\mathcal{KG}\_class\_Label_2 \leftarrow Annotate(\text{T'}.col_j, getMostFrequentClass(class\_annot))$
    $\mathcal{KG}\_candidates \leftarrow QueryEngine(\mathcal{KG}\_class\_Label_1, \mathcal{KG}\_class\_Label_2)$
    $property\_annot \leftarrow \mathcal{KG}\_candidates$;
    Annotate($\mathcal{T}'$.col$_i$,T'.$col_j$, $getBestRankedProperty(property\_annot)$)
```

The `CPA` task aims to annotate the relationship between two cells within a row by utilizing a specific property. This task follows a similar approach to the `CTA` and `CEA` tasks, employing analogous techniques and methodologies. However, there is a key distinction in the `CPA` task, as the SPARQL query is designed to select both the entity and the corresponding attributes. During the `CPA` task, the matching of properties becomes straightforward due to the prior determination of properties in the `CEA` and `CTA` task processing stages. This ensures seamless integration of the `CPA` task into the overall annotation process. The primary objective of the `CPA` task is to establish and annotate the relationship between two cells within a row, leveraging the identified properties. By leveraging the information obtained from the `CEA` and `CTA` tasks, the `CPA` task contributes to enhancing the semantic understanding and interpretation of the tabular data.

Disambiguation It is important to acknowledge that an entity within Knowledge Graphs can have multiple classes associated with it. The presence of multiple classes for an entity enriches its representation and provides a more comprehensive understanding of its semantic context within the Knowledge Graphs.

- **CTA Candidates Disambiguation**: To determine the optimal annotation for a given class or column from the available semantic annotation candidates, a selection process is employed that is based on voting and distance similarities. This involves calculating the average score of the search results, considering contributions from each column and scores obtained from distance similarity calculations. The goal is to identify the candidate feature with the highest average score, which is then chosen as the final annotation. The voting mechanism aggregates preferences from different columns, aiding in the selection of the most suitable annotation. Additionally, the consideration of distance similarities provides a measure of proximity between candidate features and the desired annotation. Through this process, we enhance the accuracy and reliability of the final annotation decision.

- **CEA Candidates Disambiguation**: To determine the optimal feature annotation among available semantic annotation candidates, a selection process is used that involves calculating the average score of the search results, taking into account both the row-based voting score and the distance similarity score. The row-based voting score reflects preferences from different rows, while the distance similarity score quantifies the similarity between candidate features and the desired annotation. By combining these scores, the candidate with the highest average score is selected as the final annotation. This process ensures alignment with row preferences and strong similarity to the desired annotation, facilitating the identification of the most appropriate and reliable feature annotation.

- **CPA Candidates Disambiguation**: In selecting the optimal annotation for a property between two columns from available semantic annotation candidates, a selection approach is used that involves calculating the average score of the search results. This calculation considers both the Match value score between cells of the two columns and the similarity score of distances. The Match value score measures the compatibility between cell values, while the similarity score evaluates the proximity between candidate properties. The candidate with the highest average score, reflecting both the Match value score and distance similarity score, is selected as the final property annotation. This process ensures that the chosen property annotation effectively captures relationships between the two columns, considering both cell value compatibility and distance similarity.

4.3 $\mathcal{KG}$_Candidates Filtering Module

The candidate annotation filtering process is facilitated by an efficient and rapid Information Retrieval technique. Once candidate annotations are identified, they are indexed and stored in a NoSQL database, specifically MongoDB. The final annotation is then determined as the result of querying this database using its integrated search engine, selecting the candidate annotation with the highest score and top rank, as outlined in lines 10 and 8 of Algorithms 1, 2, and 3, respectively [5]. MongoDB is chosen for its execution speed, scalability, and search efficiency, which contribute to the enhanced performance and effectiveness of the candidate annotation filtering process. By leveraging MongoDB's capabilities, efficient retrieval and selection of the most suitable annotations are ensured, streamlining the overall annotation workflow.

5 Kepler-aSI **performance and results**

In this section, we will present the results of Kepler-aSI for the various matching tasks in the second round of SemTab 2024[6]. These results highlight the strengths of Kepler-aSI, showcasing its encouraging performance despite the range of challenges encountered.

In Round 2, the datasets are expanded versions of those from Round 1, aiming to assess the accuracy of solutions that can scale effectively, given the common trade-off between accuracy and performance. The dataset *WikidataTables2024R2*[7] is quite similar to its Round 1 counterpart but features slight variations and includes 78,745 tables. Additionally, the datasets *tBiodivL - Large*[8] and *tBiomedL-Large*[9] are used, with *Wikidata*[10] serving as the target knowledge graph. For offline use, the March

[6] https://sem-tab-challenge.github.io/2024/tracks/accuracy-track.html

[7] https://sem-tab-challenge.github.io/2024/tracks/accuracy-track.html

[8] https://zenodo.org/records/10283083

[9] https://zenodo.org/records/10283119

[10] https://zenodo.org/records/12588085

20, 2024 dump is available, and assistance with triplestore setup can be sought from the organizers.

All datasets are organized into two data folds: training and validation. Specifically, WikidataTables comprises relational (horizontal) tables, while tBiodiv and tBiomed feature both entity (vertical) and relational (horizontal) tables. The supported tasks and their formats are as follows:

- WikidataTables: CEA, CTA, CPA
- Relational Tables in tBiomed & tBiodiv: CEA, CTA, CPA, RA, TD
- Entity Tables in tBiomed & tBiodiv: CEA, RA, TD

The target formats are:

- CEA: table name, column id, row id
- CTA: table name, column id
- CPA: table name, subject column id, object column id
- RA: table name, row id
- TD: table name

Now, we will present the results of Kepler-aSI for the various matching tasks in the second round of SemTab 2024. These results highlight the strengths of Kepler-aSI, showcasing its encouraging performance despite the range of challenges encountered.Summary of metrics for this round is in Table 1.

Table 1. Results for Round 2

	F1 Score	**Precision**	**Rank**
TBiodiv-Large-Relational-CTA	0.741	0.741	1
TBiomed-Large-Relational-CTA	0.867	0.867	1

6 Conclusion & Future Work

To summarize and conclude, we have presented in this paper the second version of our Kepler-aSI approach. Our system is participating in the challenge for the second 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.

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] Nora Abdelmageed and Sirko Schindler. “JenTab: Matching Tabular Data to Knowledge Graphs”. In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2020) co-located with the* 19th *International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 5, 2020.* Vol. 2775. CEUR Workshop Proceedings. 2020, pp. 40–49.

[2] Nora Abdelmageed et al. “Results of semtab 2022”. In: *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching* 3320 (2022).

[3] Nora Abdelmageed et al. “Semantic Web Challenge on Tabular Data to Knowledge Graph Matching”. In: *International Semantic Web Conference.* 2022.

[4] Ahmad Alobaid and Oscar Corcho. “Balancing coverage and specificity for semantic labelling of subject columns”. In: *Knowledge-Based Systems* 240 (2022), p. 108092.

[5] Wiem Baazouzi, Marouen Kachroudi, and Sami Faiz. “A Journey to Enhance Tabular Data FAIRness: From Annotation to Repair and Augmentation”. In: *2023 International Conference on Innovations in Intelligent Systems and Applications (INISTA).* IEEE. 2023, pp. 1–6.

[6] Wiem Baazouzi, Marouen Kachroudi, and Sami Faiz. “A Matching Approach to Confer Semantics over Tabular Data Based on Knowledge Graphs”. In: *Model and Data Engineering: 11th International Conference, MEDI 2022, Cairo, Egypt, November 21–24, 2022, Proceedings.* Springer. 2022, pp. 236–249.

[7] Wiem Baazouzi, Marouen Kachroudi, and Sami Faiz. “An Interactive Tool to Bootstrap Semantic Table Interpretation”. In: *Procedia Computer Science* 225 (2023), pp. 3839–3855.

[8] Wiem Baazouzi, Marouen Kachroudi, and Sami Faiz. “Kepler-aSI at SemTab 2021.” In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2021) co-located with the* 20th *International Semantic Web Conference (ISWC 2021), Virtual conference (originally planned to be in Berlin, Heidelberg), october 27, 2021.* Vol. 3103. CEUR Workshop Proceedings, 2021, pp. 54–67.

[9] Wiem Baazouzi, Marouen Kachroudi, and Sami 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* 19th *International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 5, 2020.* Vol. 2775. CEUR Workshop Proceedings. 2020, pp. 50–58.

[10] Wiem Baazouzi, Marouen Kachroudi, and Sami Faiz. “Towards an Efficient FAIRification Approach of Tabular Data with Knowledge Graph Models”. In: *Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 26th International Conference KES-2022, Verona, Italy and Virtual Event, 7-9 September 2022. Procedia Computer Science 207, Elsevier 2022.* Vol. 207. Elsevier, 2022, pp. 2727–2736.

[11] Wiem Baazouzi, Marouen Kachroudi, and Sami Faiz. “Yet Another Milestone for Kepler-aSI at SemTab 2022”. In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2022) co-located with the* 21th *International Semantic Web Conference (ISWC 2022), Virtual Event, October 23–27, 2022, Proceedings.* Springer. 2022, pp. 80–91.

[12] Steven Batt et al. “Learning Tableau: A data visualization tool”. In: *The Journal of Economic Education* 51.3-4 (2020), pp. 317–328.

[13] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. “TabEL: Entity linking in web tables”. In: *Proceedings of the* 14th *International Semantic Web Conference (ISWC), Bethlehem, PA, USA.* Springer. 2015, pp. 425–441.

[14] Yoan Chabot et al. “DAGOBAH: An End-to-End Context-Free Tabular Data Semantic Annotation System”. In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching co-located with the* 18th *International Semantic Web Conference, SemTab@ISWC 2019, Auckland, New Zealand, October 30, 2019.* Vol. 2553. CEUR Workshop Proceedings. 2019, pp. 41–48.

[15] Jiaoyan Chen et al. "Colnet: Embedding the semantics of web tables for column type prediction". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 29–36.

[16] Jiaoyan Chen et al. "Learning semantic annotations for tabular data". In: *arXiv preprint arXiv:1906.00781* (2019).

[17] Shuang Chen et al. "LinkingPark: An Integrated Approach for Semantic Table Interpretation". In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2020) co-located with the* 19^{th} *International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 5, 2020*. Vol. 2775. CEUR Workshop Proceedings. 2020, pp. 65–74.

[18] Marco Cremaschi, Roberto Avogadro, David Chieregato, et al. "MantisTable: an Automatic Approach for the Semantic Table Interpretation." In: *SemTab@ ISWC* 2019 (2019), pp. 15–24.

[19] Vincenzo Cutrona et al. "Results of SemTab 2021". In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching* 3103 (2022), pp. 1–12.

[20] Vincenzo Cutrona et al. "Results of semtab 2021". In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2021) co-located with the* 20^{th} *International Semantic Web Conference (ISWC 2021), Virtual conference (originally planned to be in Berlin, Heidelberg), october 27, 2021*. Vol. 3103. CEUR Workshop Proceedings, 2021, pp. 1–12.

[21] Dong Deng et al. "Scalable column concept determination for web tables using large knowledge bases". In: *Proceedings of the VLDB Endowment* 6.13 (2013), pp. 1606–1617.

[22] Xiang Deng et al. "Turl: Table understanding through representation learning". In: *ACM SIGMOD Record* 51.1 (2022), pp. 33–40.

[23] Sainyam Galhotra and Udayan Khurana. "Semantic search over structured data". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 3381–3384.

[24] Sainyam Galhotra et al. "Automated feature enhancement for predictive modeling using external knowledge". In: *2019 International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2019, pp. 1094–1097.

[25] Oktie Hassanzadeh et al. "Results of SemTab 2023". In: *CEUR Workshop Proceedings*. Vol. 3557. 2023, pp. 1–14.

[26] Madelon Hulsebos et al. "Sherlock: A deep learning approach to semantic data type detection". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 1500–1508.

[27] Ernesto Jiménez-Ruiz et al. "Results of semtab 2020". In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2020) co-located with the* 19^{th} *International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 5, 2020*. Vol. 2775. 2020, pp. 1–8.

[28] Ernesto Jiménez-Ruiz et al. "Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems". In: *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17*. Springer. 2020, pp. 514–530.

[29] Zhongjun Jin, Yeye He, and Surajit Chauduri. "Auto-transform: learning-to-transform by patterns". In: *Proceedings of the VLDB Endowment* 13.12 (2020), pp. 2368–2381.

[30] Sean Kandel et al. "Wrangler: Interactive visual specification of data transformation scripts". In: *Proceedings of the sigchi conference on human factors in computing systems*. 2011, pp. 3363–3372.

[31] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. "Annotating and searching web tables using entities, types and relationships". In: *Proceedings of the VLDB Endowment* 3.1-2 (2010), pp. 1338–1347.

[32] Jixiong Liu et al. "Radar Station: Using KG Embeddings for Semantic Table Interpretation and Entity Disambiguation". In: *International Semantic Web Conference*. Springer. 2022, pp. 498–515.

[33] Hiroaki Morikawa. "Semantic Table Interpretation using LOD4ALL." In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2019) co-located with the* 18th *International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 30, 2019.* CEUR Workshop Proceedings. 2019, pp. 49–56.

[34] Mark Mucchetti and Mark Mucchetti. "Google data studio". In: *BigQuery for Data Warehousing: Managed Data Analysis in the Google Cloud* (2020), pp. 401–416.

[35] Sebastian Neumaier et al. "Multi-level semantic labelling of numerical values". In: *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I 15.* Springer. 2016, pp. 428–445.

[36] Daniela Oliveira and Mathieu d'Aquin. "Adog-annotating data with ontologies and graphs". In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2019) co-located with the* 18th *International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 30, 2019.* Vol. 2775. CEUR Workshop Proceedings. 2019, pp. 40–49.

[37] BI Power. *Microsoft power platform.* 2020.

[38] Erhard Rahm and Philip A Bernstein. "A survey of approaches to automatic schema matching". In: *the VLDB Journal* 10 (2001), pp. 334–350.

[39] Bram Steenwinckel, Filip De Turck, and Femke Ongenae. "MAGIC: Mining an Augmented Graph using INK, starting from a CSV." In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2021) co-located with the* 20th *International Semantic Web Conference (ISWC 2021), Virtual conference (originally planned to be in Berlin, Heidelberg), october 27, 2021.* CEUR Workshop Proceedings. Springer-Verlag, 2021, pp. 68–78.

[40] Yoshihiko Suhara et al. "Annotating columns with pre-trained language models". In: *Proceedings of the 2022 International Conference on Management of Data.* 2022, pp. 1493–1503.

[41] Avijit Thawani et al. "Entity linking to knowledge graphs to infer column types and properties." In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2019) co-located with the* 18th *International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 30, 2019.* Vol. 2019. CEUR Workshop Proceedings. 2019, pp. 25–32.

[42] Shalini Tyagi and Ernesto Jiménez-Ruiz. "LexMa: Tabular Data to Knowledge Graph Matching using Lexical Techniques". In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2020) co-located with the* 19th *International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 5, 2020.* Vol. 2775. CEUR Workshop Proceedings. 2020, pp. 59–64.

[43] Gilles Vandewiele et al. "CVS2KG: Transforming Tabular Data into Semantic Knowledge". In: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching co-located with the* 18th *International Semantic Web Conference, SemTab@ISWC 2019, Auckland, New Zealand, October 30, 2019.* Vol. 2553. CEUR Workshop Proceedings. 2019, pp. 33–40.

[44] Daheng Wang et al. "Tcn: Table convolutional network for web table interpretation". In: *Proceedings of the Web Conference 2021.* 2021, pp. 4020–4032.

[45] Dan Zhang et al. "Sato: Contextual semantic type detection in tables". In: *arXiv preprint arXiv:1911.06311* (2019).

[46] Yi Zhang and Zachary G Ives. "Finding related tables in data lakes for interactive data science". In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data.* 2020, pp. 1951–1966.