The Journal of Supercomputing (2025) 81:1494
https://doi.org/10.1007/s11227-025-07951-y

KEPLER-ASI.V2: A BLENDED HEURISTIC FRAMEWORK FOR COMPREHENSIVE SEMANTIC TABLE INTERPRETATION

Wiem Baazouzi[1] · Marouen Kachroudi[2] · Sami Faiz[3]

Received: 27 April 2025 / Accepted: 3 October 2025

Abstract
Tabular data, frequently encountered on the Web, represents information organized in a tabular format composed of rows and columns. This format is extensively employed across various web-based contexts and data storage frameworks. Furthermore, the inherent structure of tabular data encapsulates substantial semantic information, which encourages ongoing analysis and application. Therefore, the process of deriving significant insights from structured data through semantic approaches, including ontologies or Knowledge Graphs, is typically referred to as Semantic Table Interpretation (STI) or Semantic Table Annotation. In this article, we introduce KEPLER-ASI.V2, a matching methodology designed to resolve potential semantic inconsistencies between tabular data and a Knowledge Graph. This task continues to be a formidable challenge for computational systems, necessitating additional effort for the integration of cognitive capabilities into matching algorithms. The principal aim of our approach is to devise a rapid and efficient method for the annotation of tabular data using attributes extracted from a specified Knowledge Graph. Our method integrates filtering mechanisms and text pre-processing strategies. The evaluation conducted according to the SemTab challenge has yielded promising and encouraging results.

Keywords Knowledge graph · SPARQL · Tabular data · STI

✉ Wiem Baazouzi
wiem.baazouzi@ensi-uma.tn

[1] Ecole Nationale des Sciences de l'Informatique, Laboratoire de Recherche en génIe logiciel, Application Distribuées, Systèmes décisionnels et Imagerie Intelligente, LR99ES26, Université de la Manouba, 2010 Manouba, Tunisia

[2] Laboratoire d'Informatique Programmation Algorithmique et Heuristique, LR11ES14, Faculté des Sciences de Tunis, Université de Tunis El Manar, 2092 Tunis, Tunisia

[3] Ecole Nationale d'Ingénieurs de Tunis, Laboratoire de Télédétection et Systèmes d'Information à Référence Spatiale, 99/UR/11-11, Université de Tunis El Manar, 2092 Tunis, Tunisia

Published online: 24 October 2025

1 Introduction

Within the domain of Linked Open Data, the tabular format is notably common. Furthermore, the swift advancements in web engineering highlight the pressing need to thoroughly explain the semantic information found within the columns, and sometimes even within the rows. This requirement, in certain instances, poses a challenge to the effective reuse of data, be it by domain experts or non-specialists. Within this contextual framework, efforts dedicated to ensuring the FAIRification of data [1] primarily focus on semantic metadata, which characterizes the general attributes of a given dataset. However, it is imperative to acknowledge that such descriptions, on their own, prove insufficient in guaranteeing the promised interoperability and reusability of data as envisioned by the Semantic Web project [2]. The FAIR[1] principles were essentially developed to address the needs of data management and to promote wider data sharing for enhanced reuse [3]. Comprising a set of 15 recommendations, these principles aim to render data easily (re)discoverable, accessible, interoperable, and reusable. Indeed, the process of FAIRification of data encompasses three distinct phases [1]:

(i) Pre-FAIRification: This initial phase involves the identification of the purpose of FAIRification and the comprehensive analysis of both data and metadata. This entails understanding the underlying objectives and assessing the characteristics of the (meta)data;
(ii) FAIRification: This pivotal phase consists of three essential steps, namely (i) Developing a semantic model to represent the metadata, (ii) Transforming the metadata into a machine-readable representation using the previously established semantic model, and (iii) Making the metadata accessible to both humans and machines. This step aims to enhance the discoverability, accessibility, interoperability, and reusability of data.
(iii) Post-FAIRification: Following the implementation of FAIR principles, this phase involves the evaluation of whether the objectives set in the initial phase (Pre-FAIRification) have been successfully achieved. It serves as a crucial assessment to ensure that data has indeed become more accessible, interoperable, and reusable, in alignment with the intended goals of FAIRification.

The application and consolidation of the FAIR principles[2] for web data are essential for effective data management and utilization, thereby facilitating knowledge creation through data integration, cleaning, mining, and machine learning. Consequently, the effective application of FAIR principles significantly enhances the intrinsic value of data by ensuring it is findable and accessible, while simultaneously addressing semantic ambiguities. In this context, it is crucial to highlight that effective data management is not merely a goal unto itself. Instead, it acts as a comprehensive strategy that enables the discovery and acquisition of knowledge, alongside

[1] FAIR (Findable, Accessible, Interoperable, Reusable).
[2] https://www.go-fair.org/fair-principles/.

the integration and eventual reuse of data by the stakeholder community after the data is published.

Consequently, the task of semantic annotation is viewed as a unique method of acquiring knowledge. This process involves the use of formal metadata resources, structured within a semantic framework (such as integrating one or more ontologies), which is based on leveraging semantic repositories. Recent advancements in this field indicate that tabular data are meticulously transmitted to the Web through diverse formats, with the predominant format being tabular (e.g., CSV (Comma-Separated Values)). Conversely, Web tables represent significant data sources, and enriching them with semantic details can improve applications such as Web searches, query resolution, and the development of Knowledge Bases (KB). Challenges, nevertheless, encompass the scarcity of labeled data, the task of defining or revising ontologies, the integration of existing knowledge, and the scaling of solutions. This task poses significant challenges, primarily because of incomplete or ambiguous metadata (e.g., names of tables and columns). Recent studies largely categorize into supervised methods (involving the use of annotated tables for training) [4, 5] and unsupervised approaches (employing tables devoid of data specifically for learning) [5, 6].

We introduce a comprehensive solution called KEPLER-ASI.V2 for resolving these issues, focusing on aligning tabular data with Knowledge Graphs ($\mathcal{KG}$). This system constitutes an advanced evolution of the earlier framework known as KEPLER-ASI [7–10]. The enhancements introduced extend the foundational architecture of the predecessor by incorporating a more sophisticated annotation methodology, a broader and more heterogeneous spectrum of knowledge graphs, and a refined disambiguation logic designed to improve both accuracy and scalability. From this viewpoint, it is crucial to focus on data annotation as an essential component in the analysis of tabular data [11, 12], as it enables us to deduce the significance of additional information. Subsequently, interpret the implications of the tabular data within the framework of a ($\mathcal{KG}$). The datasets employed were sourced from both Wikidata and DBPedia. More broadly, such data adhere to the triples format, comprising a subject ($\mathcal{S}$), a predicate ($\mathcal{P}$), and an object ($\mathcal{O}$). This notation guarantees semantic accessibility within data and enhances the fluidity, clarity, and dependability of all data manipulations. In recent years, there has been a growing body of work on Semantic Table Interpretation. In this setting, SemTab[3] has been developed as a project that seeks to benchmark systems focused on tabular annotation, specifically involving $\mathcal{KG}$ entities, known as table annotation. The objective of our research is to realize real-time, automated annotation of tabular data. As a result, our innovative annotation methodology is fully automated, obviating the need for initial data collection regarding entities or adherence to metadata standards. Our method is expeditious and facile to deploy, leveraging extant resources such as Wikidata and DBpedia for entity acquisition.

The rest of this paper is structured as follows. In Sect. 2, we introduce essential concepts and definitions pertinent to our field of study. Subsequently, Sect. 3

[3] https://www.cs.ox.ac.uk/isg/challenges/sem-tab/.

provides an analytical examination of the related research. Section 4 outlines our own contribution, followed by an experimental analysis in Sect. 5, culminating with the conclusions drawn in Sect. 6.

2 Preliminaries

This segment offers fundamental definitions concerning $\mathcal{KG}$s and tabular datasets. Furthermore, we introduce several notations employed consistently across this paper. Additionally, we clarify the matching tasks commonly acknowledged in STI along with their related challenges.

2.1 Key notions

2.1.1 Knowledge graphs ($\mathcal{KG}$) [13]

We process $\mathcal{KG}$s utilizing RDF syntax, portraying information through RDF triples $< \mathcal{S},\mathcal{P},\mathcal{O} >$, where $\mathcal{S}$ stands for a subject (such as a class or instance), $\mathcal{P}$ denotes a predicate (e.g., a property), and $\mathcal{O}$ signifies an object (which could be a class, instance, or data value, like text, date, or number). As per Semantic Web guidelines, RDF components (*such as* classes, properties, and instances) are identified by Uniform Resource Identifiers (URIs). Much like an ontology, a $\mathcal{KG}$ comprises a terminological component (TBox) and an assertional component (ABox). The TBox utilizes RDF Schema constructors like class subsumption and property domains. The ABox encompasses relationships among entities and definitions of semantic types. In fact, integrating an ontology with the $\mathcal{KG}$ enhances expressive capabilities without relying on RDF triplet processing, thereby facilitating the inference of new data through logical reasoning. Retrieving information from a $\mathcal{KG}$ typically involves utilizing a SPARQL endpoint or employing fuzzy matching techniques that rely on indexed lexical data linked to the entities within the $\mathcal{KG}$.

Knowledge graphs ($\mathcal{KG}$s) are typically grouped into three types based on their content: domain-specific $\mathcal{KG}$s, encyclopedic $\mathcal{KG}$s, and common sense $\mathcal{KG}$s. A domain-specific $\mathcal{KG}$ is devoted to detailing a specific area of expertise. Such a $\mathcal{KG}$ has advantages in terms of accuracy and in-depth coverage of knowledge in a particular field. It can be a very rich source to support reasoning and massive knowledge processing for specific applications. As an example, we can mention the work of Huang et al. [14] who proposed a $\mathcal{KG}$ construction model giving advice to people aiming to acquire knowledge about a healthy diet. In a different scenario, Jia et al. [15] introduced a viable technique addressing cybersecurity. Initially, they created a domain ontology that served as a foundation for constructing the cybersecurity $\mathcal{KG}$. Following this, they introduced a quintuplet model utilized to generate new insights via the path ranking algorithm. Similarly, utilizing $\mathcal{KG}$s for geological data has shown to be effective, enhancing the interconnectivity between datasets, as implemented by Zhu et al. [16]. The authors showcased the application of $\mathcal{KG}$s within an intelligent system dedicated to in-depth geological data mining. An

encyclopedic $\mathcal{KG}$ typically features a substantial size, attributed to its extensive coverage across multiple domains and its frequently open, collaborative editing process. Within this context, DBpedia [17] serves as a cornerstone of the LOD (Linked Open Data) cloud, due to its extensive interconnections with other $\mathcal{KG}$s.

2.1.2 Tabular data

In the literature we identify two main Fig. 3 of tables: layout and genuine [18]. Layout tables are an efficient way to format Web pages. The particularity of these structures is that their elements do not present any semantic coherence and are devoid of any type of semantic relationship. Their major asset is the visual organization of Web pages content in order to improve the user experience on any site. Genuine tables consist of rows and/or columns incorporating human-readable knowledge. Indeed, we consider that Genuine tables are endowed with a considerable level of syntactic and semantic consistency. The semantic contribution of Genuine tables derives from their descriptive attributes. Therefore, Authentic tables include relational information that is understandable by the machine and serves as an enhancement to the STI procedure. Conversely, Layout tables are crucial for visual presentation, yet they lack meaningful semantic relationships between their cells. Consequently, they cannot support knowledge extraction and interpretation tasks. Indeed, this relational characteristic constitutes in itself an added value, since it emphasizes the topology of the semantic links between the cells of a given Genuine table. In what follows, we dissect these details in order to fully understand the contribution and importance of tabular data with a strong relational structure. Thus, a relational table has structures in which each row or possibly column describes a specific entity. While the corresponding columns or possibly rows represent attributes that describe the entity. The arrangement of entities and their attributes in the relational table indicates the orientation of their reading and manipulation, i.e., horizontally or vertically. Relational tables can have a header labeling the first row. The authors [19, 20] report that the orientation aspect emphasizes the direction of the relationships inside a table. In practice, knowing the meaning of the relationships within a table facilitates its interpretation, for example, to annotate a subject you must start by reading all of its attributes. If we consider a horizontal table, the subjects would obviously be described horizontally: each line describes a different subject. By analogy, if we consider a vertical table, the subjects would be described vertically, in other words, each column describes a different subject. Having elucidated these basics, we can synthesize the above as follows: Suppose T represents a two-dimensional table, comprising an arranged sequence of N rows and M columns, as illustrated in Fig. 1.

M_j represents a row of the table for $j = 1, 2, \ldots, M$, while N_i denotes a column of the table for $i = 1, 2, \ldots, N$. The intersection of a row M_j and a column N_i is designated as $c_{i,j}$, which is the value contained in the cell $T_{i,j}$. The table's contents can comprise various data types, such as strings, dates, floats, and numbers. Thus, the following structures will be considered: Target Table (S): M N, Subject Cell: $T_{(i,0)}$ for $i = 1, 2, \ldots, N$, and Object Cell: $T_{(i,j)}$ for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$.

2.2 STI tasks formulation

In the previous subsection, we identified our field of study through the typing of the different forms of tabular data and the $\mathcal{KG}$s commonly used to perform the annotation. In the following, we will list the STI tasks as well as their technical specificities. The annotation task is characterized by the array elements that need annotation and the kinds of candidates, such as individuals, concepts, or properties within the $\mathcal{KG}$. As stated by Semtab[4] (the initiative integrating the leading systems in this area), the work on STI is categorized into three primary tasks, as illustrated in Fig. 2:

Column-type annotation (CTA) This task seeks to align a column with an entity type belonging to a $\mathcal{KG}$. The fastidious handling of the CTA task comes down to identifying the appropriate type in a relatively complex hierarchical type structure. To put it differently, an entity might possess various types depicted in cycle graphs similar to Wikidata. The underlying principle is this: the type chosen for a specific column should effectively represent the individuals within it while providing the most information possible. A highly specific type might not adequately cover all the values in a column, thereby affecting the precision of annotation methods.
Cell-entity annotation (CEA) This process is also referred to as *Entity Linking*. At this step, STI techniques strive to associate a cell with an entity from a $\mathcal{KG}$. As shown in Fig. 2, the cell with the content "Austria" corresponds to the Austrian national football team on Wikidata (`Q163534`).
Columns-property annotation (CPA) The objective of this task is to label a pair of columns in a relational table with a specific attribute. For instance, in Fig. 2, the section highlighted in green connects the final column to the numerical figures of another column, representing the score and/or points accrued by each country during the group phase. This property is represented on Wikidata by the code (`P1351`) and the label "*number of points/goals/set scored*".

Besides the most prominent annotation tasks cited earlier, the literature also mentions less common types, such as "Topic Annotation," which is designed to tag an entire table with a concept or entity from the target $\mathcal{KG}$. This type of annotation is alternatively called “Context Annotation.” For instance, in Fig. 2, we observe that the whole table is related to the entity "*UEFA Euro 2008*", as represented in Wikidata by (`Q241864`). Moreover, Row-to-Instance annotation associates a complete row of a relational table with an entity from a $\mathcal{KG}$. Each row is handled as an entity, considered the subject of that line. To illustrate, Fig. 2 highlights the fourth row in purple, identifying the match within its context, denoted by the Wikidata code (`Q17754883`). Some tasks are complementary, and the completion of one (even partially) can contribute to the initiation of another task. The following section reviews the most relevant related works regarding STI.

[4] https://www.cs.ox.ac.uk/isg/challenges/sem-tab/.

Fig. 1 Generic shape of tabular data at a glance

$$
\begin{array}{c} \\ Row_1 \\ \\ Row_j \\ \\ Row_M \end{array}
\begin{array}{c}
\begin{array}{ccccc} Col_0 & & Col_i & & Col_N \end{array} \\
\begin{pmatrix}
S_{1,0} & \cdots & \cdots & \cdots & S_{1,N} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
S_{j,0} & \cdots & S_{j,i} & \cdots & S_{j,N} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
S_{M,0} & \cdots & \cdots & \cdots & S_{M,N}
\end{pmatrix}
\end{array}
$$

3 Scrutiny of related work

This section examines the literature by evaluating multiple contributions, with a focus on CTA, CEA, and CPA tasks. Subsequently, we analyze these methods from distinct angles: assets, deficiencies, and the influence of the examined table elements and/or the $\mathcal{KG}$ framework on performance metrics. Several studies have addressed the STI problem, differing in the techniques utilized and the approaches employed.

3.1 Overview and main trends

Table 1 summarizes that STI systems use various strategies to address CEA, CTA, and CPA tasks. Approaches range from lexical/statistical methods to embedding-based representations. Some systems combine search engines with similarity measures or use modular pipelines. Recent methods incorporate deep learning and contextual modeling, trading complexity for accuracy. Strengths include efficiency, robustness, and precision, while limitations involve reliance on external resources, sensitivity to thresholds, and scalability issues. The list of STI-related works that we have drawn up is not exhaustive, but it is a list, through which we seek to cover all trends in terms of strategies and techniques. Indeed, according to our review and the methods' specificities, it would be possible to identify two large preponderant families, namely: heuristic and deep learning based, as depicted by Fig. 3.

The literature on Semantic Table Interpretation (STI) distinguishes two main families of approaches: heuristic-based methods and deep learning-based methods [21]. Each family can be further subdivided into more specific paradigms, as detailed below.

Heuristic-based approaches Heuristic approaches rely on simple rules, similarity measures, or majority voting strategies. They are typically lightweight and do not require complex training phases, which makes them efficient for small datasets or specific use cases. Two main subcategories can be identified:

Lookup-based approaches These approaches involve the direct retrieval of prospective entities, categories, or associations from a specific Knowledge Graph ($\mathcal{KG}$), like DBpedia or Wikidata. Typically, they use techniques such as string similarity or lexical matching to connect table mentions to $\mathcal{KG}$ entries.

Iterative approaches These methods improve initial annotations through successive refinement steps. They exploit interdependencies between tasks (e.g.,

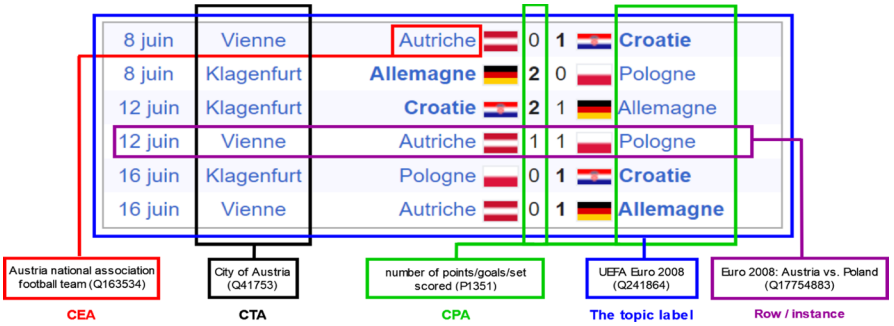

Fig. 2 An illustrative example of the most known STI tasks (concrete case of a sports event described and archived on Wikidata UEFA Euro 2008)

between cell annotation and column type annotation) and apply multiple rounds of disambiguation to ensure global consistency.

Heuristic approaches are easy to implement and computationally efficient, but they often struggle with ambiguous mentions and show limited coverage when applied to heterogeneous data.

Deep learning-based approaches Deep learning approaches leverage neural architectures to automatically learn representations of tabular data and knowledge graphs. They require large-scale training data but usually achieve superior performance by capturing semantic dependencies more effectively. Two main paradigms exist:

$\mathcal{KG}$-oriented modeling These methods focus on learning embeddings directly from the structure of the target $\mathcal{KG}$. Entities, relations, and types are represented as dense vectors, which are then used to guide the annotation process.
Table-oriented modeling These methods focus on encoding the internal structure of tables (rows, columns, headers, and contextual metadata). Using neural models, they exploit correlations within the table itself and benefit from large corpora of tabular data to learn robust, general-purpose models. Deep learning approaches provide strong robustness to ambiguity and heterogeneity but require substantial computational resources and annotated data for training. This classification provides a clear framework to analyze existing contributions, as each STI method can be mapped to one of these categories or subcategories.

3.2 STI approaches

ADOG [45] is a lookup method that utilizes aggregation of string similarities, the frequency of property appearances, and the normalized scoring from the Elasticsearch tool to align DBpedia entities in the CEA task. It incorporates score

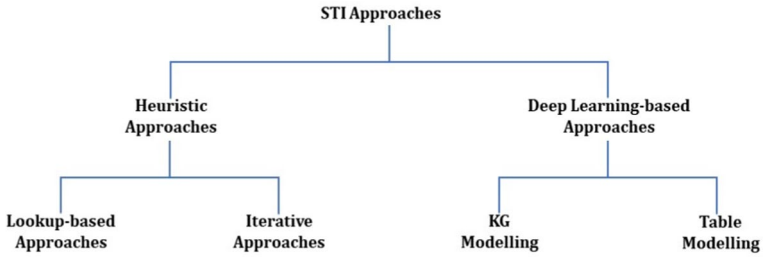

Fig. 3 Classification of STI approaches main families and classes

aggregation with IDF-based metrics, and ArangoDB[5] is employed for indexing DBpedia elements. The system applies Levenshtein distance and TF-IDF for candidate selection and exploits class/property frequency to derive CTA and CPA annotations. Tabularisi [23] adopts a statistical approach based on TF-IDF to rank candidate entities for CEA. Each candidate is encoded as a binary feature vector indicating the presence or absence of descriptive properties. Candidate scores combine TF-IDF, Levenshtein similarity, and word similarity between labels. Aggregation can be hand-crafted or learned through a two-layer neural network. CTA is computed by a brute-force search in the $\mathcal{KG}$ class hierarchy, while CPA relies on identifying frequent inter-column relations. Magic [46] introduces INK embeddings to represent attributes, values, and contextual table information. CEA is achieved by matching embeddings between cells and $\mathcal{KG}$ entities, while CTA and CPA exploit embedding aggregation at the column level. The system identifies a *key column* to retrieve candidates via public endpoints, propagating them to other cells in the row. Limitations include API restrictions, spelling variations, and synonym handling. Alobaid et al. [25] propose a heuristic approach that leverages DBpedia to perform the three tasks (CEA, CTA, CPA). Their method integrates semantic similarity measures and contextual evidence for improving candidate ranking. Kepler-aSI [26] extends the heuristic iterative paradigm by jointly modeling surface similarity, contextual evidence, and global coherence across the table. It leverages DBpedia and Wikidata, consistently participating in SemTab 2021–2024, and achieves robust improvements across CEA, CTA, and CPA. TorchicTab [27] combines heuristic rules with representation learning to enhance entity disambiguation. Using DBpedia and Wikidata as resources, the system improves candidate ranking for CEA and propagates type/property assignments for CTA and CPA. Parmar et al. [28] propose a Wikidata-based method that performs CEA, CTA, and CPA. Their iterative strategy refines entity linking using contextual signals derived from table rows and columns. CSV2KG [47] applies a six-phase pipeline where seed annotations are iteratively refined. Similarity is used to match candidate labels, column types are deduced, and co-occurrence patterns help derive relationships (CPA). Corrections to header cells occur via property annotations, leading to new column types being inferred from the adjusted cells. MTab [48] performs entity linking, type prediction,

[5] https://www.arangodb.com.

Table 1 Summary table of the reviewed methods

Approaches			Annotation tasks			$\mathcal{KG}$	Data source	Year
Class		Algorithm	CEA	CTA	CPA			
Heuristic	Lookup-based	ADOG [22]	✓	✓	✓	DBpedia	SemTab2019	2019
		Tabularisi [23]	✓	✓	✓	DBpedia	T2D,VizNet	2019
		Magic [24]	✓	✓	✓	DBpedia,Wikidata	SemTab2021	2021
		Alobaid et al. [25]		✓		DBpedia	SemTab2021,T2D	2022
		Kepler-aSI [26]	✓	✓	✓	DBpedia,Wikidata	SemTab2021-2024	2023
		TorchicTab [27]	✓	✓	✓	DBpedia,Wikidata	SemTab2023	2023
		Parmar et al. [28]	✓	✓		Wikidata	SemTab2024	2024
	Iterative	CSV2KG [29]	✓	✓	✓	DBpedia	SemTab2019	2019
		MTab [30]	✓	✓	✓	DBpedia	SemTab2019-2021	2021
		LinkingPark [31]	✓	✓	✓	DBpedia	SemTab2019-2022	2022
		DAGOBAH-SL [32]	✓	✓	✓	DBpedia,Wikidata	SemTab2020-2022	2022
		MantisTable [33]	✓	✓	✓	DBpedia,Wikidata	SemTab2020-2022	2022
		KGCODE-Tab [34]	✓	✓	✓	DBpedia,Wikidata	SemTab2022	2022
		JenTab [35]	✓	✓	✓	DBpedia,Wikidata	SemTab2020-2023	2024
		Kepler-aSI [10]	✓	✓	✓	DBpedia,Wikidata	SemTab2021-2024	2024

Table 1 (continued)

Approaches			Annotation tasks			$\mathcal{KG}$	Data source	Year
Class		Algorithm	CEA	CTA	CPA			
Deep learning	$\mathcal{KG}$ modeling	Vasilis et al. [12]	✓			Wikidata	T2D, Wikipedia	2017
		Biswas et al. [36]				DBpedia	Custom Wikipedia	2018
		DAGOBAH-Em [37]	✓	✓		DBpedia, Wikidata	SemTab2019	2019
	Table modeling	Sherlock [38]		✓		DBpedia	T2D, VizNet	2019
		Sato [39]		✓		DBpedia	VizNet	2019
		Guo et al. [40]		✓	✓	DBpedia	T2Dv2	2020
		Singh et al. [41]			✓	DBpedia	T2Dv2	2021
		DREIFLUSS [42]		✓		DBpedia	Custom Semtab2023	2023
		IDLab [43]		✓		Wikidata	Wikidata	2024
		Bikim et al. [44]	✓	✓	✓	Wikidata	Wikidata	2024

and relation extraction in an iterative manner. Candidate entities are scored by combining lexical similarity with contextual features, and global optimization ensures consistency of CEA, CTA, and CPA. LinkingPark [49] integrates entity and property linking in a joint pipeline. Entities are first identified, then refined through disambiguation and property link detection, while inter-row relationships are inferred using property characteristics. DAGOBAH [37] is implemented as a modular system with sequential complementary tools. It identifies semantic relations between tables and $\mathcal{KG}$s, enriches $\mathcal{KG}$s with new triples, and generates metadata. Candidate annotations are extracted via SPARQL queries and refined with contextual features. MantisTable [50] classifies columns into entity, literal, or subject types, retrieves candidates with SPARQL queries, and disambiguates entities row-wise. CTA is performed through a type graph hierarchy, where scores are adjusted by ontological distance from the root, while CPA uses majority voting. KGCODD-Tab [34] addresses the three annotation tasks using DBpedia and Wikidata. It exploits contextual constraints and ontology information for candidate filtering and ranking. JenTab [51] consists of nine processing modules for CEA, CTA, and CPA. It combines initial candidate generation with contextual refinement, confidence-based filtering, and fallback strategies for missing annotations, ensuring robustness even with sparse input signals. Vasilis et al. [12] and Biswas et al. [36] propose $\mathcal{KG}$ modeling approaches that rely on Wikidata and Wikipedia embeddings to enhance annotation accuracy. DAGOBAH-Embeddings [37] extends the DAGOBAH framework by exploiting embedding-based clustering (using TransE) to group candidate entities. Clusters with high coverage are selected to perform CEA and CTA. Deep learning-based methods include Sherlock [38], which applies feature engineering and neural networks for column type prediction (CTA), Sato [39], which combines learned features with $\mathcal{KG}$-based signals, Guo et al. [40] and Singh et al. [52], which integrate table representation learning with $\mathcal{KG}$ information, and DREPTLUSS [42], IDLab [43], and Bikim et al. [44], which focus on Wikidata-based STI with advanced neural architectures.

3.3 Synthesis and discussion

In the family of methods with a heuristic aspect, we note that each system has an intrinsic algorithmic organization, without resorting to learning techniques. In this type of method, the annotation tasks are based on similarity measures, majority voting or even probabilistic processing tools. The family of heuristic approaches is divided into two other groups: lookup-based and iterative. Lookup-based approaches take as input an initial set of candidate entities, which is usually obtained by a certain search service. As summarized in Table 2, this is the case for the ADOG [45] and Tabularisi [23] employs methods to handle the three annotation tasks: CTA, CEA, and CPA—utilizing Dbpedia as background knowledge. Following the creation of a candidate list through this search procedure, these methods evaluate candidates by assigning scores based on various metrics applied to the table components, such as the cells and the type of the column. As for the iterative approaches, such as LinkingPark [53] and JenTab [54–56], they generally have a search system

Table 2 Comparison of semantic table interpretation systems

System	Main techniques	Strengths	Limitations
ADOG [22]	String similarity aggregation, TF-IDF weighting, Elasticsearch scoring	Combines multiple similarity measures with IDF scores	Limited coverage, strong dependence on string similarity
Tabularisi [23]	TF-IDF, word similarity, Levenshtein distance, neural aggregation	Combines lexical and neural signals	High feature complexity, brute-force CTA search
Magic [24]	INK embeddings, per-column similarity, key column search	Embedding-based matching for robustness	Limited by API, synonym and spelling issues
Kepler-aSI [26]	Contextual similarity, column-based scoring, coherence	Exploits inter-column relations for disambiguation	Sensitive to noise in context
TorchicTab [27]	Neural scoring functions, tabular embeddings	Deep context-aware table encoding	Requires large training data
Parmar et al. [28]	Wikidata-based entity linking, iterative refinement	Adapted for Wikidata-specific schema	Limited portability to other $\mathcal{KG}$s
CSV2KG [29]	Multi-phase annotation, iterative refinement	Progressive disambiguation improves accuracy	Complex multi-step process
MTab [30]	Multilingual matching, iterative entity linking	Multilingual support	Limited by DBpedia coverage
LinkingPark [31]	Candidate generation + disambiguation, property inference	Joint entity and property linking	Pipeline complexity
DAGOBAH-SL [32]	Candidate selection, iterative disambiguation	Scalable and modular	Heavy reliance on $\mathcal{KG}$ queries
MantisTable [33]	Column categorization, SPARQL querying, majority voting	Leverages ontology hierarchy	Sensitive to thresholds and errors propagation
KGCODE-Tab [34]	$\mathcal{KG}$-guided table annotation, semantic validation	$\mathcal{KG}$ consistency checking	Dependent on $\mathcal{KG}$ quality
Kepler-aSI [10]	Coherence-based scoring, contextual embeddings	Strong contextual disambiguation	Higher complexity
DAGOBAH-Em [37]	TransE embeddings, $\mathcal{KG}$-based clustering	Embedding consistency improves CTA	Sensitive to noisy candidates
Sherlock [38]	Pre-trained embeddings, convolutional neural networks	Strong generalization across schemas	Limited to column type annotation
Sato [39]	Deep column embeddings, probabilistic classification	Exploits large training corpus	Limited to CTA

Table 2 (continued)

System	Main techniques	Strengths	Limitations
Guo et al. [40]	Table serialization, deep learning for relation extraction	Captures inter-column dependencies	Model complexity
Singh et al. [41]	Neural property matching, relation inference	Improves CPA with deep context	Limited to property annotation
IDLab [43]	Graph embeddings, large-scale table annotation	Good scalability for big data	$\mathcal{KG}$ dependency
Bikim et al. [44]	Contextual neural scoring, attention mechanisms	Strong coherence modeling	Data-hungry model

as their base, reinforced by an additional phase of multi-task disambiguation which makes it possible to rectify the candidate entities classification. These iterative disambiguation techniques have a remarkable impact on improving the performance of any approach. The second family of methods is based on deep learning. Indeed, we have noticed in recent years that access to large volumes of information, combined with robust computing resources, has given success to this technique, which draws on these aforementioned resources to carry out the learning phase. Thus, some STI works have adopted deep learning several times, revealing two trends: $\mathcal{KG}$ Modeling and Tables Modeling. The fundamental concept behind $\mathcal{KG}$ Modeling involves utilizing $\mathcal{KG}$ embedding methods to represent entities and their associations within a designated vector space. As evidenced in Table 2, this approach is exemplified by DAGOBAH Embeddings [37] and Radar Station [57]. In this part, STI approaches hypothesize that entities belonging to the same column are likely to be semantically similar. Indeed, in the embedding space, they must also be close to each other. The approaches like DUDUO [58] and TCN [59] focus more on defining the context in the Table Modeling axis. Indeed, they seek to provide a contextualized representation of the table's basic elements, which is learned using deep neural networks or language models such as BERT. The text within the table undergoes thorough examination to assess potential interactions, both within the table itself and between different tables. We also notice that the methods are always in search of maximum coverage over tables elements considered as input, whereas this proves in certain cases to be very costly in terms of execution time and/or resources.

Table 2 provides a comparative overview of Semantic Table Interpretation (STI) systems. It highlights the main techniques, strengths, and limitations of each approach.

This also leads us to a perpetual problem which is the trade-off between effectiveness and efficiency: *how to set up an efficient system with the least cost (or an optimal cost) ?*. On the other hand, systems that adopt a heuristic approach rely on matching techniques, which are efficient as long as there are no problems of noise or incompleteness of tabular data. In addition, while some methods circumvent this dependency by deploying deep learning techniques, they remain strictly constrained by data availability (to perform the training phase) as well as high-performance computing resources. Finally, it is appropriate to mention the key role that $\mathcal{KG}$s play in the annotation phase and more specifically encyclopedic $\mathcal{KG}$s like Wikidata or DBpedia. Indeed, a good mastery of such $\mathcal{KG}$s and their structures added to a good querying technicality and filtering gives access to a mine of information (once properly valued) which can be the best basis for triggering the annotation process. Despite the diversity of strategies and techniques, we also note that all methods share a uniform and recurrent modus operandi, which inherits good data science practices (everything concerning their processing, storage, exploitation, etc.). The ultimate goal of each STI system is to perform annotation tasks while preserving maximum semantics. This global process is commonly summarized as follows:

- *Pre-processing* The annotation quality provided by an STI system depends on the input data. Indeed, a data pre-analysis is essential during the first step of a successful STI system. In this preliminary analysis stage, normalizing the for-

mat allows the original data to be converted into a format that an STI system can read. This task also aims to clean up, whether numeric or textual, the compatibility between the different data sources. In addition, data analysis tends to extract as much as possible information encapsulated in the table before starting the annotation phase.

- *Candidates generation* This phase relies on a task-dependent list for array elements: CTA, CEA, or CPA. The task's goal directs processing: CTA begins with creating a candidate types list while CEA starts with a candidate entities list.
- *Navigability over table cells* Annotation of tabular data involves interpreting rows, columns, entire tables, or metadata. Rows detail an entity's attributes, while columns highlight entities in the same column. Combining these interpretations refines candidate selection for a table cell, enhancing STI system precision.
- *Disambiguation* This process, known as iterative disambiguation, involves valuing each task's output to perform subsequent tasks. It is widely used in heuristic approaches to reduce search space and enhance STI systems' accuracy in detecting correct annotations.

3.4 Contextual positioning and principal contributions

Unlike previous systems such as Kepler-aSI [60] or JenTab [55], DAGOBAH [37] and MTab [61], where tables are processed independently without sharing information, our approach explicitly exploits inter-table dependencies. It addresses table heterogeneity by enforcing type consistency across columns and enabling relational inference across multiple tables. We assume that the first row n_1 serves as the header, with the first cell of each column ($c_{1,j}$) defining the column header, which standardizes the representation. Although real-world tables may have more complex structures (missing headers, multi-row headers, irregular layouts), this simplification facilitates processing and allows robust alignment of entities, types, and relations. Overall, the proposed system improves STI by leveraging cross-table information, integrating contextual signals at both the column and relation levels, and applying knowledge graph guided scoring functions to enhance annotation accuracy in CEA, CTA, and CPA tasks. The next section introduces our contribution, namely a novel STI system called Kepler-aSI.v2, designed to support tabular data annotation by combining established and emerging techniques.

4 The proposed new STI system

When implementing Kepler-aSI.v2 (Kepler as a Semantic Interpreter, Version 2), we opt for a heuristic approach that trades off effectiveness and efficiency. For our system, the natural language barrier does not pose an obstacle thanks to its integration of cross-lingual methods and enhanced queries for Wikidata (or a similar $\mathcal{KG}$). These characteristics cohere with the FAIR principles introduced in Sect. 1. The overall workflow is illustrated in Fig. 4.

4.1 Module 1: Pre-processing

Initially, we undertake pre-processing of the table cells because of issues with text encoding in the data. This step is essential to reduce the likelihood of errors in subsequent stages. We carry out the following procedures.

4.1.1 Data type detection

At this stage, we identify the data type for each cell in the table, determining if they contain numbers or text :

- Concerning numerical values: We conduct data type identification for each cell value $c_{i,j}$ within a table. Duckling,[6] a tool that evaluates text against 13 different data types, is employed for this purpose. The tool utilizes a context-free probabilistic grammar system. Different data types are associated with numeric identifiers. Examples include an amount of money, credit card numbers, distances, durations, scalars, ordinals, telephone numbers, quantities, temperatures, times, volumes, or distinctive tags like email addresses or URLs. When a data type is identified, it is designated as a numeric cell. If no data type is determined, the cell is considered a text cell.
- Concerning text data, we apply entity type detection to each table cell $c_{i,j}$ using SpaCy[7] models. In the process known as SpaCyNER (SpaCy Named Entity Recognition), entity types are determined via a pre-trained deep-learning model. This model is utilized on our dataset to interpret cell values, yielding a classification of typed named entities.
- Using Regextypes,[8] we validate and support data type detection results from Duckling and SpaCyNER. Our regular expressions cover various units and entities like area, currency, mass, speed, and more, verifying if table cells contain numeric or text values.

4.1.2 Natural language detection

Considering that tabular data is written in multiple natural languages, not just one, it is essential to develop a method to address and resolve this limitation. In fact, for language identification, we utilized the langrid[9] library, enabling us to detect 26 languages within our dataset. The langrid library functions as an independent tool for language recognition. It is engineered to accommodate a broad spectrum of languages. In practical terms, it is sufficient to extract a portion of textual data and process it through the langrid in order to ascertain the language identification.

[6] https://github.com/facebook/duckling.
[7] https://spacy.io/api/entityrecognizer.
[8] https://docs.python.org/3/howto/regex.html.
[9] https://github.com/openlangrid.

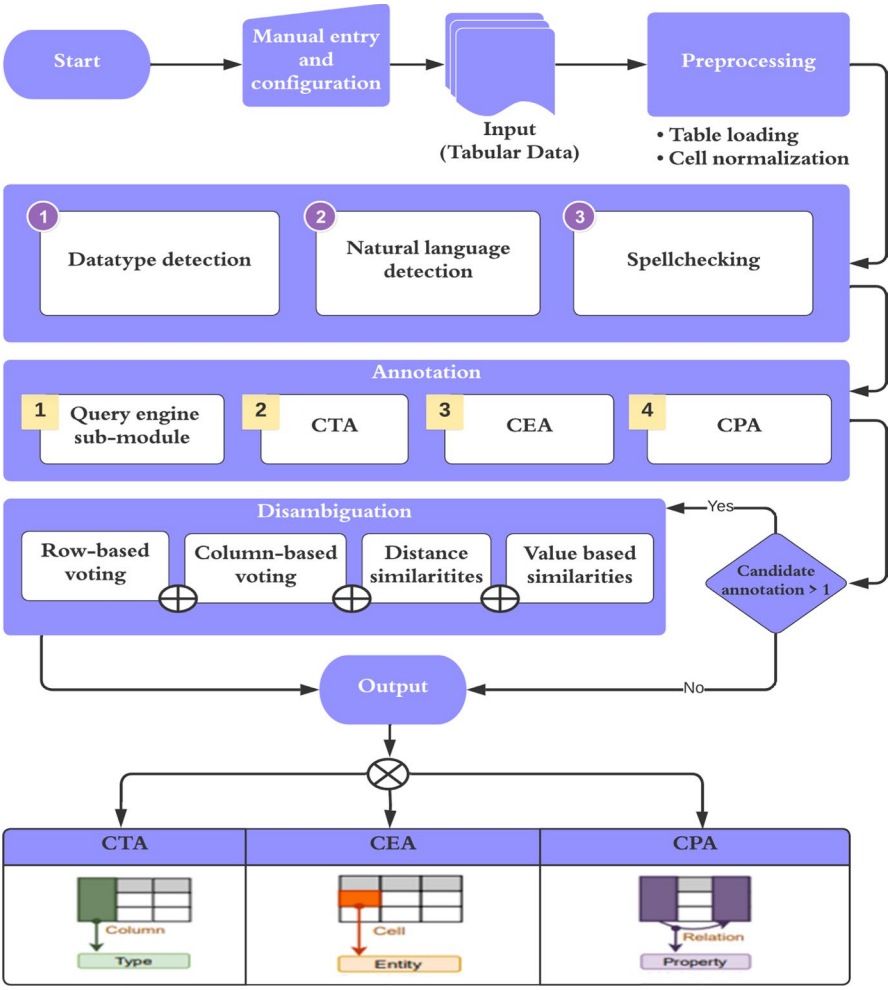

Fig. 4 The architecture of KEPLER-ASI.v2 system

4.1.3 Spellchecking

According to the literature [41], spell-checkers are considered essential linguistic tools in natural language processing (NLP). They are widely employed in diverse applications such as information extraction, proofreading, information retrieval, social media, and search engines.

In the pre-processing phase of our work, we conducted a comparative study of several spell-checking approaches and libraries, including TextBlob,[10] Spark NLP,[11] Gurunudi,[12] Wikipedia API,[13] PySpellChecker,[14] SerpAPI,[15] Autocorrect,[16] ftfy,[17] and PyGrammalecte.[18] A comparative overview of some of these libraries is provided in Table 3. Following a comprehensive evaluation and consideration of compatibility issues with other pre-processing components, we have elected to employ *TextBlob* and PySpellChecker as the principal tools for the purpose of spelling correction and the purification of noisy textual data within the analyzed tables.

4.2 Module 2: Annotation

Upon the completion of the various pre-processing steps, the phase of annotating the tabular data may proceed.

4.2.1 Query engine sub-module

This module is central to the annotation process. We use a parameterized SPARQL query to extract annotations from $\mathcal{KG}$s. Initially, the switcher evaluates the context and runs the appropriate query.

4.2.2 Assigning a semantic type to a column (CTA)

Each element is labeled with a tag, facilitating semantic recognition. The CTA task utilizes suitable APIs to search for an item based on its description. In our approach, the data accumulated for a specific entity includes an instance list (denoted by the `instanceOf` primitive and retrievable using P31 code), the subclass of (denoted by `subclassOf` and accessible via P279 code), and overlaps (indicated by `partOf` accessible with P361 code). With this information, we are equipped to handle the CTA task through a SPARQL query, as indicated in Listing 1.

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

Listing 1: A SPARQL query example aimed at extracting CTA candidates from Wikidata.

[10] https://textblob.readthedocs.io/en/dev/.
[11] https://nlp.johnsnowlabs.com/.
[12] https://github.com/guruyuga/gurunudi.
[13] https://wikipedia.readthedocs.io/en/latest/code.html.
[14] https://github.com/barrust/pyspellchecker.
[15] https://serpapi.com/spell-check.
[16] https://pypi.org/project/autocorrect/.
[17] https://pypi.org/project/ftfy/
[18] https://pypi.org/project/pygrammalecte/.

Table 3 Spell-checking libraries trends

Name	Category	Strengths/limitations
TextBlob	NLP	Spelling correction, easy-to-use
Spark NLP	NLP	Pre-trained models, text analysis, multilingual support
Gurunudi	NLP	Pre-trained models, text analysis, easy-to-use, multilingual
Wikipedia API	Search engine	Search with suggestions, easy-to-use, unlimited access
PySpellChecker	Spell checking	Simple spell-checking algorithm, no pre-trained models, easy-to-use
SerpAPI	Search engine	Limited access for free users
Autocorrect	Spell checking	Simple and fast, but less accurate; no contextual analysis
ftfy	Text cleaning	Focused on fixing Unicode/encoding errors; not a true spell-checker
PyGrammalecte	Grammar/spell checking	Designed for French text; grammar + spelling correction, limited multilingual support

In accordance with Algorithm 1 from KEPLER-ASI.V2, the goal is to label each entity column with components from the relevant $\mathcal{KG}$, which were determined as its type in the pre-processing step. The SPARQL query serves as our method of inquiry, informed by entity data that dictates the selection of each data type, considering they represent a list of instances (P31), subclasses (P279), or class segments (P361). The outcome of a SPARQL query might typically involve a single type; however, occasionally, it encompasses multiple types, necessitating a disambiguation process. The annotation of a specific column c within table $\mathcal{T}$ is designated as $\mathcal{A}^c$. Each candidate derived from the query is systematically indexed using swift and effective Information Retrieval (IR) methodologies. In fact, once annotations are detected, they are catalogued and stored in a NoSQL database, specifically MongoDB. The concluding annotation, seen as the outcome of the matching procedure, is then identified by querying this database using its built-in search engine. We chose MongoDB to arrange a nested local dictionary and due to its notable advantage in execution speed, owing to its processing strengths such as scalability and search efficiency. This index is represented by $\mathcal{I}^c$. Specifically, a preserved annotation stems from the equation specified in Eq. 1 (where the "*" symbol denotes the general nature of the row dimension in the table, i.e., focusing on the j^{th} column's cells):

$$\mathcal{A}^c(\mathcal{T}[*, j]) = argmax(f(Label, \mathcal{I}^c)) \tag{1}$$

Doing so, this expression returns the label having the highest frequency (most number of occurrences over the index). If it turns out that there are several candidates satisfying this criterion, we carry out a disambiguation process. The global annotation, i.e., of all the table columns, is the union of the annotations of each of the columns belonging to the aforementioned table, as formulated by Eq. 2:

$$\mathcal{A}^c{}_{\mathcal{T}} = \biguplus \mathcal{A}^c(\mathcal{T}[*, j]) \tag{2}$$

To enhance the annotation process in KEPLER-ASI.V2 over [7–9], the approach queries three knowledge graphs simultaneously, boosting the chances of obtaining the most appropriate and semantically accurate entity annotation. We enhance the joint use of *DBpedia* and *Wikidata* to *YAGO*,[19] a multilingual knowledge graph sourced from Wikipedia, WordNet, and GeoNames, improving disambiguation and semantic representation, which were lacking in prior work [7–9, 32, 48, 62]. Candidates undergo selection and normalization, including URI filtering, type hierarchy harmonization, and duplicate elimination, to ensure broad and reliable coverage. Inter-$\mathcal{KG}$ alignment links DBpedia, Wikidata, and YAGO entities using lexical matching, label normalization, and unique identifiers like URIs `owl:sameAs`. YAGO's semantic relations from WordNet help capture synonymy and hierarchy. Then, fusion and disambiguation combine candidate sets from the graphs $\mathcal{KG}$, retaining the most coherent entity based on inter-$\mathcal{KG}$ agreement, type granularity, and table structure context. This strategy produces precise, semantically rich annotations with improved multilingual coverage and detailed typing, following the steps in Algorithm 1. An empty set `class_annot` is initialized for each table column $\mathcal{T}$ to store the candidate class. Each cell is then analyzed by extracting its label (lines 5 6). During candidate generation (lines 7 9), *DBpedia*, *Wikidata*, and *YAGO* identify entities or classes for the label. Candidate sets are then merged (line 10) to enhance semantic coverage and reduce ambiguity through multilingual and inter-$\mathcal{KG}$ information. The disambiguation phase (lines 11 15) selects a class for each column: a single remaining candidate is retained, or a CTA-specific function chooses the best class. The algorithm outputs the annotated table $\mathcal{T}\simeq$ (line 16), ensuring semantic coherence and leveraging multiple knowledge graphs and disambiguation mechanisms.

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

Listing 2: SPARQL query for CEA task.

4.2.3 Assigning a cell to a $\mathcal{KG}$ entity (CEA)

Similarly to the CTA task, the CEA task can be executed using a SPARQL query, as demonstrated in Listing 2.Algorithm 1: CTA task Kepler-aSI.v2

[19] https://yago-knowledge.org/.

Algorithm 1: CTA task Kepler-aSI.v2

Input: Table $\mathcal{T}$
Output: Annotated Table $\mathcal{T}'$
$i \leftarrow 0$
$j \leftarrow 0$
while $col_i \in \mathcal{T}$ **do**
  $class\_annot \leftarrow \emptyset$
  while $cell_{i,j} \in col_i$ **do**
    $Label \leftarrow cell.expressionValue$
    $\mathcal{L}_{DB} \leftarrow QueryEngine(Label, DBpedia)$
    $\mathcal{L}_{WD} \leftarrow QueryEngine(Label, Wikidata)$
    $\mathcal{L}_{Yago} \leftarrow QueryEngine(Label, Yago)$
    $\mathcal{KG}\_candidates \leftarrow AlignAndFuse(\mathcal{L}_{DB}, \mathcal{L}_{WD}, \mathcal{L}_{Yago})$
    if $\mathcal{KG}\_candidates.size = 1$ **then**
      $class\_annot \leftarrow \mathcal{KG}\_candidates$
    else
      if $\mathcal{KG}\_candidates.size > 1$ **then**
        $class\_annot \leftarrow CTA\_disambiguation(\mathcal{KG}\_candidates)$
Retourner $\mathcal{T}'$

The CEA task, as outlined in Algorithm 2 in KEPLER-ASI.V2, seeks to tag the cells of a specified table to a particular entity found on Wikidata or potentially another knowledge graph ($\mathcal{KG}$). Additionally, if the cells in question are part of columns that were previously labeled during the CTA task, their outcome can be adapted accordingly. This procedure mirrors that of the CTA task. Should the SPARQL query yield a sole candidate, this candidate is selected as the annotation. For cases with multiple candidates, further processing is required to resolve ambiguities. Formally, the treatment is represented by Eq. 3:

$$\mathcal{A}^r(\mathcal{T}[i,*]) = argmax(f(Label, \mathcal{I}^r)) \tag{3}$$

In Eq. 3, $\mathcal{A}^r(\mathcal{T}[i,*])$ denotes the assignment function associated with relation r, applied to the i-th row of the data matrix $\mathcal{T}$, i.e., the feature vector corresponding to instance i. The variable $\mathcal{I}^r$ represents the information or representation related to relation r, while *Label* indicates the supervised label. The function $f(\text{Label}, \mathcal{I}^r)$ defines a scoring or compatibility measure between the label and the relational representation. Finally, arg max is the operator that selects the argument maximizing the function f. The index encapsulating the labels of all the considered cells is denoted $\mathcal{I}^r$ (similarly the symbol "*" is used to express the columns genericity in the table, i.e., to target the cells of the i^{th} row). By adopting the same approach as that of the previous task, we retain the label with the highest frequency.

While disambiguation processing is performed if we have candidates with the same score. The global annotation, i.e., of all the concerned cells of the table is the union of all the annotations of each of the cells, as formulated by Eq. 4:

$$\mathcal{A}^r{}_{\mathcal{T}} = \biguplus \mathcal{A}^r(\mathcal{T}[i,*]) \tag{4}$$

In Eq. 4, $\mathcal{A}^r{}_{\mathcal{T}}$ denotes the global assignment with respect to relation r over the data matrix $\mathcal{T}$. The operator $\biguplus$ represents the disjoint union (or aggregation) over instances. Each term $\mathcal{A}^r(\mathcal{T}[i,*])$ corresponds to the assignment of relation r with respect to the i-th instance vector, i.e., the i-th row of $\mathcal{T}$.

The procedure follows the structured workflow of the CTA task in KEPLER-ASI.V2, with cell-level adjustments. We use multiple knowledge graphs to enhance candidate annotation identification, addressing the limitations of single knowledge graph reliance cited in prior works [7–9, 62]. Refer to Algorithm 2. For each row row_j of the table (line 4), the algorithm iterates over each cell $cell_{i,j}$ (line 5). It extracts the cell label (line 6) and queries it against *DBpedia*, *Wikidata*, and *YAGO* to generate candidate sets (lines 7 9). These sets are aligned and merged (line 10) using multilingual labels and `owl:sameAs` links, reducing redundancy and enhancing semantic coverage. A disambiguation step follows (lines 11 14). If a single candidate remains (line 12), it is assigned to the cell. A CEA-specific disambiguation function selects the most suitable entity based on inter-$\mathcal{KG}$ agreement, type granularity, and row-level context. This method, combining joint candidate generation, inter-$\mathcal{KG}$ alignment, and contextual disambiguation, leads to more accurate and enriched cell annotations, improving on past methods. After processing all cells, the algorithm outputs the annotated table $\mathcal{T}_{\simeq}$.

Algorithm 2: CEA task Kepler-aSI.v2

Input: Table $\mathcal{T}$
Output: Annotated Table $\mathcal{T}'$
1 $i \leftarrow 0$
2 $j \leftarrow 0$
3 **foreach** $row_j \in \mathcal{T}$ **do**
4 **foreach** $cell_{i,j} \in row_j$ **do**
5 $Label \leftarrow cell.expressionValue$
6 $\mathcal{L}_{DB} \leftarrow QueryEngine(Label, DBpedia)$
7 $\mathcal{L}_{WD} \leftarrow QueryEngine(Label, Wikidata)$
8 $\mathcal{L}_{YAGO} \leftarrow QueryEngine(Label, YAGO)$
9 $\mathcal{KG}\_candidates \leftarrow AlignAndFuse(\mathcal{L}_{DB}, \mathcal{L}_{WD}, \mathcal{L}_{YAGO})$
10 **if** $|\mathcal{KG}\_candidates| = 1$ **then**
11 $entity\_annot \leftarrow \mathcal{KG}\_candidates$
12 **else**
13 $entity\_annot \leftarrow CEA\_disambiguation(\mathcal{KG}\_candidates, context\_signals)$
14 **Retour** $\mathcal{T}'$

4.2.4 Assigning a property to a $\mathcal{KG}$ entity (CPA)

A key distinction in the CPA task is that the SPARQL query is required to retrieve both the entity and its associated attributes, as seen in Listing 3.

```
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. }
}
```

Listing 3: SPARQL query for CPA task.

Upon annotating the values and types associated with each considered entity, we proceed to determine the relationships between two cells on the same row through a property, using a SPARQL query as outlined in Algorithm 3. In fact, the CPA task involves marking the connection between two cells in a specific row using a property. This task is similarly carried out in a manner akin to the CTA and CEA tasks.

Formally, this relational property is modeled by Eq. 5:

$$\mathcal{A}_{\mathcal{T}}^{c_{n,m}}(\mathcal{T}[i,j]) = \mathcal{R}_p(\mathcal{A}_{\mathcal{T}}^{c_n}, \mathcal{A}_{\mathcal{T}}^{c_m}) \tag{5}$$

In Eq. 5, $\mathcal{A}_{\mathcal{T}}^{c_{n,m}}(\mathcal{T}[i,j])$ denotes the joint assignment with respect to the pair of classes (c_n, c_m) for the entry located at position (i, j) in the data matrix $\mathcal{T}$. The terms $\mathcal{A}_{\mathcal{T}}^{c_n}$ and $\mathcal{A}_{\mathcal{T}}^{c_m}$ correspond to the global assignments associated with classes c_n and c_m, respectively. The operator $\mathcal{R}_p(\cdot,\cdot)$ represents a relational function or composition rule that combines the two class-based assignments according to a predefined relation p. Where: $n \neq m$; and $\mathcal{R}_p$ is the property linking two classes labeling already two annotated cells. Identifying the properties is straightforward because they have already been detected during the CEA and CTA task processing. In the following, we provide a real data fragment example to clarify the various procedures and treatments utilized in our proposed KEPLER-ASI.V2 system.

In the CPA task KEPLER-ASI.V2, we adapt the principles from CTA and CEA for annotating table column properties. This method, integrating joint candidate generation, inter-$\mathcal{KG}$ alignment, and contextual disambiguation, marks a significant enhancement over KEPLER-ASI [7–9, 37]. It ensures precise and meaningful property annotations, enhancing the annotation pipeline from the CTA and CEA tasks. Algorithm 3 explains the process. For each column pair (col_i, col_j) in the table $i \neq j$ (line 4), an empty set `property_annot` is created to store candidate properties. For each cell pair $(cell_1, cell_2)$ in these columns (line 5), the labels $Label_1$ and $Label_2$ are extracted (lines 6 7).Label pairs are queried against *DBpedia*, *Wikidata*, and *YAGO* to generate candidate properties (lines 8 10). The sets are aligned and merged (line 11) using unique identifiers and inter-$\mathcal{KG}$ links such as `owl:sameAs` to reduce redundancy and improve coverage. A disambiguation step follows (lines 12 15): if only one candidate remains, it is selected (line 13); otherwise, a CPA-specific function chooses the most relevant property based on inter-$\mathcal{KG}$ consistency and contextual signals from the table structure (lines 14 15). After processing all column pairs, the algorithm returns the annotated table $\mathcal{T}_{\simeq}$ (line 16), enriched with precise semantic relations between columns.

Algorithm 3: CPA task Kepler-aSI.v2

Input: Table $\mathcal{T}$
Output: Annotated table $\mathcal{T}'$ with column properties
1 $i \leftarrow 0$
2 $j \leftarrow 0$
3 **foreach** $(col_i, col_j) \in \mathcal{T}, i \neq j$ **do**
4 $\quad property\_annot \leftarrow \emptyset$
5 $\quad$ **foreach** $(cell_1, cell_2) \in (col_i, col_j)$ **do**
6 $\quad\quad Label_1 \leftarrow cell_1.expressionValue$
7 $\quad\quad Label_2 \leftarrow cell_2.expressionValue$
8 $\quad\quad \mathcal{L}_{DB} \leftarrow QueryEngine(Label_1, Label_2, DBpedia)$
9 $\quad\quad \mathcal{L}_{WD} \leftarrow QueryEngine(Label_1, Label_2, Wikidata)$
10 $\quad\quad \mathcal{L}_{YAGO} \leftarrow QueryEngine(Label_1, Label_2, YAGO)$
11 $\quad\quad \mathcal{KG}\_candidates \leftarrow AlignAndFuse(\mathcal{L}_{DB}, \mathcal{L}_{WD}, \mathcal{L}_{YAGO})$
12 $\quad\quad$ **if** $|\mathcal{KG}\_candidates| = 1$ **then**
13 $\quad\quad\quad property\_annot \leftarrow \mathcal{KG}\_candidates$
14 $\quad\quad$ **else**
15 $\quad\quad\quad property\_annot \leftarrow CPA\_disambiguation(\mathcal{KG}\_candidates, context\_signals)$
16 **Retour** $\mathcal{T}'$

4.2.5 Disambiguation

It is worth mentioning that an entity may be depicted in Knowledge Graphs using multiple classes. In DBpedia, for instance, dbr:Barack_Obama is represented by the classes dbo:Person, dbo:Politician, and dbo:President. Comparatively, in Wikidata, Barack Obama is associated with the classes owl:Q5, owl:Q82955, and owl:Q30461 respectively. This suggests that Knowledge Graphs allow for multiple methods of annotating entities, showcasing the diverse facets, roles, and characteristics linked to them. Indeed, utilizing several classes to characterize an entity enriches its depiction and allows for a more comprehensive understanding of its semantic context in Knowledge Graphs. The vast amount of data present at the representation level can complicate the task of identifying appropriate candidates for the annotations produced by our $\mathcal{KG}$ method. To tackle this issue, we have developed a targeted method known as disambiguation, aimed directly at the previously introduced annotation tasks.

4.2.5.1 CTA\candidates\disambiguation We employ a selection methodology utilizing voting and distance likenesses to determine the best annotation for a specific class or column. This task entails calculating the average weighted score for the candidate features by taking into account contributions from each pertinent column, along with the scores derived from distance similarity calculations, as described in Algorithm 4.

The CTA algorithm KEPLER-ASI.V2 contextually disambiguates table cells, efficiently handling simple cases and using a multi-step process for ambiguous ones. For each column col_i and cell $cell_{i,j}$, it generates candidate entities from the knowledge graph based on cell labels (lines 5 6). For multiple candidates, a structured disambiguation process follows (lines 7 9). A column-based voting mechanism captures contextual

preferences (line 7), followed by a distance-based function prioritizing semantically closer candidates (line 8). Voting and distance scores are combined into a weighted score (line 9). The candidate possessing the top score is selected as the best cell annotation (line 10). The algorithm outputs the completely annotated table $\mathcal{T}'$ (line 11).

Algorithm 4: CTA disambiguation in Kepler-aSI.v2

Input: Table $\mathcal{T}$, Knowledge Graph $\mathcal{KG}$
Output: Annotated Table $\mathcal{T}'$
1 $i \leftarrow 0$
2 $j \leftarrow 0$
3 **while** $col_i \in \mathcal{T}$ **do**
4 **while** $cell_{i,j} \in col_i$ **do**
5 $Label \leftarrow cell_{i,j}.expressionValue$
6 $\mathcal{E}(cell_{i,j}) \leftarrow CandidateGenerator(Label, \mathcal{KG})$
 `// Step 1: Column-based voting to capture row/column context`
7 $\mathcal{E}_{vote} \leftarrow ColumnBasedVoting(\mathcal{E}(cell_{i,j}), col_i)$
 `// Step 2: Distance-based similarity to prioritize semantically close candidates`
8 $\mathcal{E}_{dist} \leftarrow CalculDistance(\mathcal{E}_{vote}, \mathcal{KG})$
 `// Step 3: Compute combined score integrating voting and distance`
9 $\mathcal{E}_{weighted} \leftarrow WeightedAverage(\mathcal{E}_{vote}, \mathcal{E}_{dist})$
 `// Step 4: Select candidate with highest combined score as final annotation`
10 $class\_annot \leftarrow \arg \max_{e \in \mathcal{E}_{weighted}} Score(e)$
11 **return** $\mathcal{T}'$

By combining *voting, distance similarity, local scoring, global coherence, and multilingual $\mathcal{KG}$ integration*, this approach constitutes a significant improvement over previous systems [7–9, 48], providing robust, context-aware, and semantically informed selection of the most appropriate column type annotation. For example, when annotating "Barack Obama" using Wikidata, the candidate classes with initial scores are: `Person - 0.90`, `Politician - 0.85`, `President - 0.80`, `OfficeHolder - 0.75` and `Political Leader - 0.70`. The results are as follows :

Step 1: voting-based

1. Person : 0.90
2. Politician : 0.85
3. President : 0.80
4. OfficeHolder : 0.75
5. Political Leader : 0.70

Step 2: Distance-based similarity

1. Person : 0.60
2. Politician : 0.70
3. President : 0.50
4. OfficeHolder : 0.40
5. Political Leader : 0.80

Step 3: Weighted average scoring

1. Person : $(0.90 + 0.60)/2 = 0.75$
2. Politician : $(0.85 + 0.70)/2 = 0.775$
3. President : $(0.80 + 0.50)/2 = 0.65$
4. OfficeHolder : $(0.75 + 0.40)/2 = 0.575$
5. Political Leader : $(0.70 + 0.80)/2 = 0.75$

Step 4: Local & Global Scoring Integration

1. Person : $0.624 + 0.10 = 0.724$
2. Politician : $0.702 + 0.35 = 1.052$
3. President : $0.808 + 0.45 = 1.258$
4. OfficeHolder : $0.623 + 0.25 = 0.873$
5. Political Leader : $0.700 + 0.40 = 1.100$

At this stage, each table cell has candidate entities scored by local text match (e.g., "Barack Obama" matches "President") and global coherence with neighboring cells. The cell is annotated with the candidate that achieves the top combined score, ensuring both local precision and global coherence. This top candidate, *President*, boasts the highest weighted score overall and is thus designated to this cell (*c.f.*, Table 4). Step 4 involves selecting the candidate balancing local similarity and global coherence. Aggregating relevant scores ensures each cell gets an annotation accurate locally and consistent globally. The weighted combination of scores determines the final choice.

4.2.5.2 CEA candidates disambiguation To identify the optimal annotation for a feature among several candidate options, we use a selection process based on average scores computed from search results. This process takes into account both the semantic relevance of the candidates and their contextual proximity to the desired annotation. To determine the optimal annotation for a table cell from multiple candidate entities, KEPLER-ASI.V2 employs a row-aware selection process that combines *row-based voting* and *distance-based similarity*. Each candidate receives a weighted score reflecting both its semantic relevance and its contextual proximity to the intended annotation. The complete procedure is outlined in Algorithm 5.

Algorithm 5: CEA Disambiguation in Kepler-aSI.v2

Input: Table $\mathcal{T}$, Knowledge Graph $\mathcal{KG}$
Output: Optimal Annotations $\mathcal{KG}\_candidates$

$i \leftarrow 0$
$j \leftarrow 0$
while $col_i \in \mathcal{T}$ **do**
  while $cell_{i,j} \in col_i$ **do**
    $Label \leftarrow cell_{i,j}.expressionValue$
    $\mathcal{E}(cell_{i,j}) \leftarrow CandidateGenerator(Label, \mathcal{KG})$
    // Step 1: Row-based voting to capture column context
    $\mathcal{E}\_vote \leftarrow RowBasedVoting(\mathcal{E}(cell_{i,j}), col_i)$
    // Step 2: Distance-based similarity to prioritize semantically close candidates
    $\mathcal{E}\_dist \leftarrow CalculDistance(\mathcal{E}\_vote, \mathcal{KG})$
    // Step 3: Compute weighted score combining voting and distance
    $\mathcal{E}\_weighted \leftarrow WeightedAverage(\mathcal{E}\_vote, \mathcal{E}\_dist)$
    // Step 4: Select candidate with highest combined score as final annotation
    $class\_annot \leftarrow \arg \max_{e \in \mathcal{E}\_weighted} Score(e)$
return $\mathcal{KG}\_candidates$

Example: When annotating the entity “Republican” using a knowledge graph such as Wikidata, the initial candidate entities and their base scores are: `Republican - 0.30`, `Republican River - 0.23`, `Klamath Republican - 0.20` and `Republican, Arkansas - 0.18`. The results are as follows :

Table 4 Summary of scores for each candidate (CTA) KEPLER-ASI.V2

Candidate	Voting	Distance	Local	Global	Final weighted
Person	0.90	0.60	0.624	0.10	–
Politician	0.85	0.70	0.702	0.35	–
President	0.80	0.50	0.808	0.45	Selected
OfficeHolder	0.75	0.40	0.623	0.25	–
Political Leader	0.70	0.80	0.700	0.40	–

Step 1: Row-Based Voting

1. Republican : 0.30
2. Republican River : 0.23
3. Klamath Republican : 0.20
4. Republican, Arkansas : 0.18

Step 2: Distance-based similarity

1. Republican : 1.0
2. Republican River : 0.7
3. Klamath Republican : 0.8
4. Republican, Arkansas : 0.9

Step 3: Weighted average scoring

1. Person : $(0.90 + 0.60)/2 = 0.75$
2. Politician : $(0.85 + 0.70)/2 = 0.775$
3. President : $(0.80 + 0.50)/2 = 0.65$
4. OfficeHolder : $(0.75 + 0.40)/2 = 0.575$
5. Political Leader : $(0.70 + 0.80)/2 = 0.75$

Step 4: Local & Global Scoring Integration

1. Person : $0.624 + 0.10 = 0.724$
2. Politician : $0.702 + 0.35 = 1.052$
3. President : $0.808 + 0.45 = 1.258$
4. OfficeHolder : $0.623 + 0.25 = 0.873$
5. Political Leader : $0.700 + 0.40 = 1.100$

In order to disambiguate the mention Republican against potential candidate entities in Wikidata, we applied a multi-step scoring procedure combining row-based voting, distance-based similarity, and local global integration. As summarized in Table 5, the initial candidates (Republican, Republican River, Klamath Republican, and Republican, Arkansas) received base scores from row-based voting and were subsequently refined using semantic distance metrics.

The local and global scores were then integrated to obtain the final weighted ranking. Among the candidates, Klamath Republican achieved the highest final score (1.258), making it the most probable entity for annotation in this context. This stepwise approach is consistent with recent work on entity linking that emphasizes the integration of local and global context to improve disambiguation accuracy. This row-aware voting mechanism, combined with distance-based similarity, ensures that each table cell is annotated with the candidate that best aligns with both the contextual preferences of the rows and the semantic proximity to the target entity. The use of a such technique in KEPLER-ASI.V2 represents a significant improvement over the original KEPLER-ASI system [7–9] or other ones in the same register [62, 63], providing more precise and semantically rich Cell Entity Annotations (CEA).

4.2.5.3 CPA candidates disambiguation In KEPLER-ASI.V2, the optimal property connecting two table columns is determined using a cell-pair-aware selection process that combines cell compatibility scoring with distance-based similarity. Each

Table 5 Summary of scores for each candidate (CEA in KEPLER-ASI.V2)

Candidate	Row-based vote	Distance	Local	Global	Final weighted
Republican	0.30	1.00	0.624	0.10	–
Republican River	0.23	0.70	0.702	0.35	–
Klamath Republican	0.20	0.80	0.808	0.45	Selected
Republican, Arkansas	0.18	0.90	0.623	0.25	–

candidate property is assigned a weighted score that captures both its alignment with the values of the paired cells and its semantic closeness to the target property. The complete procedure is outlined in Algorithm 6.

Algorithm 6: CPA disambiguation Kepler-aSI.v2

Input: Table $\mathcal{T}$, Candidate Properties $\mathcal{KG}\_candidatesCPA$
Output: Optimal Property Annotations $\mathcal{KG}\_candidates$

$i \leftarrow 0$
$j \leftarrow 0$
while $cell_1 \in col_i$ *and* $cell_2 \in col_j$ **do**
  while $Label_1$ *and* $Label_2 \in \mathcal{KG}\_candidatesCPA$ **do**
    // Step 1: Compute cell compatibility between the paired cells
    $\mathcal{E}\_compat \leftarrow CellCompatibilityScore(Label_1, Label_2)$
    // Step 2: Distance-based similarity to prioritize semantically close properties
    $\mathcal{E}\_dist \leftarrow CalculDistance(\mathcal{E}\_compat)$
    // Step 3: Weighted average scoring combining compatibility and distance
    $\mathcal{E}\_weighted \leftarrow WeightedAverage(\mathcal{E}\_compat, \mathcal{E}\_dist)$
    // Step 4: Select candidate with highest weighted score as final property
    $property\_annot \leftarrow \arg \max_{p \in \mathcal{E}\_weighted} Score(p)$
return $\mathcal{KG}\_candidates$

For example, consider annotating a property linking a column of people to a column of years, e.g., "George Washington" and "1789". Applying the same disambiguation principle mentioned above, we obtain the following result:

Step 1: Cell Compatibility Score

1. Start of term : 0.20
2. Death year : 0.15
3. Birth year : 0.35
4. End of term : 0.10

Step 2: Distance-based similarity

1. Start of term : 0.20
2. Death year : 0.60
3. Birth year : 1.00
4. End of term : 0.70

Step 3: Weighted average scoring

1. Start of term : $(0.20 + 0.20)/2 = 0.20$
2. Death year : $(0.15 + 0.60)/2 = 0.375$
3. Birth year : $(0.35 + 1.00)/2 = 0.675$
4. End of term : $(0.10 + 0.70)/2 = 0.40$

Step 4: Local & Global Scoring Integration

1. Start of term : $0.20 + 0.10 = 0.30$
2. Death year : $0.375 + 0.15 = 0.525$
3. Birth year : $0.675 + 0.50 = 1.175$
4. End of term : $0.40 + 0.20 = 0.60$

Table 6 summarizes the scores assigned to each candidate for establishing the relation between "George Washington" and "1789".

The evaluation process includes the Cell Compatibility Score (CPA), which measures the direct match between cells, the Distance score, reflecting similarity based on numeric or temporal values, and the Local and Global scores, which integrate context-specific and broader contextual information, respectively. The Final Weighted column combines these measures to identify the most relevant candidate. In this case, the Birth year was selected as the optimal match, despite other candidates having high scores in certain metrics, highlighting the importance of a weighted integration of multiple criteria for robust decision-making. This cell-pair-aware and distance-informed approach ensures the selected property reflects both alignment between column values and semantic relevance, completing the pipeline alongside CTA and CEA for consistent entity, type, and property annotation.

5 Evaluation

In this section, we present the experimental study conducted to evaluate our contribution, namely the Kepler-aSI.v2 system. First, we present the test bases, as well as their technical characteristics which highlight the different identified challenges. Next, we review the evaluation metrics that allow us to qualitatively examine our method. Indeed, the performance of Kepler-aSI.v2 is presented through three successive participations in the Semtab 2021, Semtab 2022 and Semtab 2023 campaigns dedicated to the STI systems evaluation. During the development of Kepler-aSI.v2, we adopted an incremental approach, which allowed us to introduce successive improvements. Thus, the evaluation of our participation in Semtab 2021 is divided into two stages: the intra-method perspective (to investigate the internal structure sensitivity of our system), and the inter-methods perspective to have a positioning of our system in relation to the other participating systems. Our experimental study ends with a discussion that provides an objective assessment of the Kepler-aSI.v2 performance. In the context of the SemTab 2022 and 2023 challenges, we conducted a rigorous investigation into the responsiveness of our system to the dynamic nature of test dataset.

5.1 Data features and statistics

The Semtab 2021 Accuracy track included 3 rounds, where different $\mathcal{KG}$s were used during the different rounds, namely: DBpedia[20] (version 2016–10), Wikidata[21] and Schema.org.[22] Indeed, this version on the challenge employed a variety of dataset to assess participating systems' performance across distinct difficulty levels. These dataset included: (*i*) Tough Tables (2T): Evaluating robustness against challenging

[20] http://downloads.dbpedia.org/wiki-archive/.
[21] https://zenodo.org/record/6153449.
[22] https://gittables.github.io/downloads/schema20210528.pkl.

Table 6 Summary of scores for each candidate relation between "George Washington" and "1789"

Candidate	Cell compatibility score	Distance	Local	Global	Final weighted
Start of term	0.20	0.20	0.20	0.10	–
Death year	0.15	0.60	0.375	0.15	–
Birth year	0.35	1.00	0.675	0.50	Selected
End of term	0.10	0.70	0.40	0.20	–

tables; (*ii*) BioTable: Focusing on complex molecular biology data (largest number of rows); (*iii*) Automatically Generated (AG): The largest dataset, comprising tables generated programmatically via SPARQL queries; (*iv*) BiodivTab: Featuring real-world tables from biodiversity research; (v) GitTables: A large-scale corpus of relational tables extracted from GitHub. The SemTab 2022 challenge adopted the same $\mathcal{KG}$ as 2021, with a key update: a more recent Dbpedia version (early 2022). Test dataset included: (*i*) Tough Tables and HardTables[23] (HT): Evaluating performance on challenging tables; (*ii*) BiodivTab[24]: Adapted real-world biodiversity research tables, targeting Dbpedia instead of Wikidata as the $\mathcal{KG}$. (*iii*) GitTables[25]: A larger corpus than 2021 for training data-driven methods, with a curated subset for benchmarking column type detection. The SemTab 2023 challenge introduced two tracks: Accuracy and dataset. The Accuracy Track assessed system performance across diverse dataset, tasks, and Knowledge Graphs with varying difficulty levels. Unlike previous iterations (2019–2021; AIcrowd-facilitated), participants submitted solutions through a dedicated form, with evaluation occurring at the end of each round. Like SemTab 2022, partial ground truth data (training/validation sets) was provided for local methodology evaluation. Four distinct dataset groups were employed across the two rounds. The SemTab 2023 challenge employed four diverse dataset: (*i*) WikidataTables[26]: This dataset (9,917 tables) features realistic, SPARQL-generated tables with high ambiguity for feature columns, aiming to assess system robustness under challenging conditions; (*ii*) tFood[27]: This domain-specific dataset (11,588 tables) focuses on food-related data, encompassing horizontal relational tables and entity tables. Ground truth mappings and a novel Topic Detection task were provided; (*iii*) SOTAB[28]: Designed for Column Type Annotation (CTA) and Column Property Annotation (CPA), this dataset leverages down-sampled WDC Schema.org tables with varying vocabulary size to create a graduated difficulty level; (*iv*) CQA[29]: This Wikary-based dataset (844 tables) utilizes Wikipedia tables annotated

[23] https://doi.org/10.5281/zenodo.7419275.
[24] https://doi.org/10.5281/zenodo.7319654.
[25] https://zenodo.org/record/7091019.
[26] https://doi.org/10.5281/zenodo.8393535.
[27] https://doi.org/10.5281/zenodo.7828163.
[28] https://doi.org/10.5281/zenodo.8422037.
[29] https://doi.org/10.5281/zenodo.8398347.

with Wikidata qualifiers. The class imbalance due to qualifier asymmetry presents a challenge for machine learning models.

5.2 Evaluation metrics

In this experimental study, as part of our participation in the evaluation campaign, we adhere to the Semtab evaluation framework. Here, the annotation targets, which include cells, columns, and columns—are predetermined, and any unnecessary annotations are disregarded. During the two Semtab 2021 and Semtab 2022 campaigns, and for the various annotation tasks, standard Precision, Recall and F1-Score metrics were used, as they are expressed by Eqs. 6, 7 and 8. This is applicable for Dbpedia and Schema.org.

$$\text{Precision} = \frac{|correct\_annotations|}{|submitted\_annotations|} \tag{6}$$

In Eq. 6, *Precision* measures the proportion of correctly identified annotations among all submitted annotations. The term $|correct\_annotations|$ denotes the number of annotations that match the ground truth, while $|submitted\_annotations|$ represents the total number of annotations produced by the system.

$$\text{Recall} = \frac{|correct\_annotations|}{|ground\_trouth\_annotations|} \tag{7}$$

In Eq. 7, *Recall* quantifies the proportion of correctly identified annotations with respect to all the annotations present in the ground truth. The term $|correct\_annotations|$ denotes the number of annotations correctly retrieved by the system, whereas $|ground\_truth\_annotations|$ corresponds to the total number of annotations defined in the ground truth.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

In Eq. 8, the *F1-score* is defined as the harmonic mean of Precision and Recall. This metric offers a balanced evaluation by taking into account both the system's proficiency in returning only pertinent annotations (Precision) and its capability to capture all relevant annotations (Recall). The numerator $2 \times Precision \times Recall$ emphasizes the joint contribution of both measures, while the denominator $Precision + Recall$ normalizes the score.In cases where target annotations refer to specific target cells for CEA, to particular target columns for CTA, and to designated pairs of target columns for CPA.According to track organizers, an annotation is correct if it is part of the *ground truth* set, as target cells often have multiple ground truth annotations due to redirects and `same-as` links in $\mathcal{KG}$s. The organizers implemented two precision and recall approximations for the CTA task [64, 65], factoring in Wikidata's structural granularity. These adjustments include partially correct annotations, such as ancestors or descendants of the *ground truth* classes. A

correctness score *cscore* evaluates CTA annotations based on their distance from the *ground truth* classes in the type hierarchy.

5.3 Performance

In this section, we will review and comment on the performance of our contribution, the KEPLER-ASI.V2 method. As mentioned, this evaluation is carried out through three consecutive participations in the Semtab campaign, in its successive versions 2021, 2022, and 2023.

5.3.1 Results for Semtab 2021 test bases

To better understand the technical contributions of our method, we present two intra- and inter-method evaluation scenarios. On the one hand, we show the sensitivity of our method to the details of data used, and on the other hand, we provide a comparative study with leading methods participating in the evaluation campaign, to position our method.

5.3.1.1 Intra-method perspective The test sets challenge method robustness by inserting erroneous data. Orthographic errors in tables can harm annotation results, and flawed SPARQL query labels may yield no results, reducing candidates and lowering response quality. At this level, we would like to recall that we deploy two complementary tools to address this challenge, namely TextBlob and Pyspellchecker. During the first round of Semtab 2021, we implemented this processing. As shown in Table 7, by introducing the correction to our KEPLER-ASI.V2 system, it marked an improvement of between 11% and 26% in terms of F1 measures and a gain of between 2% and 20% in terms of precision.

The second challenge is processing test data in 26 languages. We utilize a translation tool to adjust SPARQL query language labels, guided by Natural Language Detection for identification. This approach improved F1-Measure by 18–20% and Precision by 17% across all test cases. Moreover, querying Knowledge Graphs often results in multiple annotation candidates, introducing the challenge of disambiguation. Without it, annotation quality and system metrics suffer due to reduced accuracy in identifying correct annotations. Disambiguation techniques improve our system's F1-Measure by 2–15%. This highlights the impact of each treatment on our KEPLER-ASI.V2 system's performance. Our optimal setup, proven in Round 1 of Semtab 2021, incorporates three treatments and is maintained for later tests. Table 8 shows that using multiple knowledge graphs with YAGO consistently boosts task performance.

Leveraging *Multi-KG* broadens semantic coverage and aids complex correspondences, especially in CEA and CTA. Adding YAGO provides extra semantic knowledge, leading to systematic improvements in precision and F1-score. In the CTA-DBP task, the F1-score improves from 0.35 *Multi-KG* to 0.39 *Multi-KG+YAGO*, and in CEA-DBP from 0.42 to 0.47. Likewise, improvements occur in CTA-WD (0.72 → to 0.76) and CEA-WD (0.62 → to 0.67). In challenging scenarios like HARD

Table 7 Kepler-aSI.v2's performance on the SemTab 2021 from an intra-method perspective (F1 and Pr denote F1-measure and precision respectively)

	CEA-DB		CTA-DB		CEA-WD		CTA-WD	
	F1	Pr	F1	Pr	F1	Pr	F1	Pr
☐ Correction	0.215	0.684	0.114	0.333	0.336	0.740	0.246	0.480
☑ Correction	0.355	0.720	0.240	0.520	0.520	0.770	0.390	0.520
☐ Translation	0.355	0.720	0.240	0.520	0.520	0.770	0.390	0.520
☑ Translation	0.510	0.800	0.420	0.650	0.720	0.790	0.560	0.660
☐ Disambiguation	0.510	0.800	0.420	0.650	0.720	0.790	0.560	0.660
☑ Disambiguation	0.620	0.830	0.510	0.690	0.810	0.800	0.730	0.760
Final Evaluation	0.625	0.835	0.515	0.695	0.815	0.805	0.735	0.765

and domain-specific tasks (BIO, BioDiv), integrating YAGO further boosts performance, with F1-scores consistently over 0.90 in BIO and HARD, and rising from 0.54 → to 0.58 in BioDiv. These results show that integrating heterogeneous knowledge graphs with YAGO enhances semantic coverage and correspondence resolution in benchmark tasks.

5.3.1.2 Inter-method perspective In Semtab 2021, KEPLER-ASI participated with 5 other systems as outlined by Table 9, and successfully processed almost all of the proposed cases, with the exception of 4 cases.

5.3.2 Results for Semtab 2022 test bases

The evaluation of KEPLER-ASI on the SemTab 2022 datasets highlights the impact of integrating multiple knowledge graphs and external enrichment. *Multi-KG vs baseline:* Using Multi-KG alone already provides strong performance across standard tasks, with F1-scores of 0.96 for CTA, 0.97 for CEA, and 0.99 for CPA. On harder datasets, Multi-KG demonstrates resilience: for instance, Hard-CTA-WD and Hard-CEA-WD reach F1 scores of 0.92 and 0.90 respectively, while even more challenging cases such as Tough-CTA-WD and Tough-CEA-WD achieve 0.50 and 0.53, respectively. *Multi-KG+YAGO vs Multi-KG:* Adding YAGO as an external source further boosts results. Standard tasks improve slightly, with F1 increases of 1.04% for CTA (0.96–0.97), 1.03% for CEA (0.97–0.98), and 0.5% for CPA (0.99–0.995). Hard datasets also benefit: Hard-CTA-WD gains +1.09% (0.92–0.93) and Hard-CEA-WD +2.25% (0.89–0.91). More significant improvements occur on the toughest cases, such as Tough-CTA-DBP (+8.57%) and Tough-CEA-DBP (+6%). Biodiversity datasets also show clear gains, with F1 for Biodiv-CTA-DBP rising from 0.70 to 0.72 (+2.9%) and for Biodiv-CEA-DBP from 0.68 to 0.71 (+4.4%). Table 10 shows that extending from Multi-KG to Multi-KG+YAGO consistently improves F1 and Precision, especially on challenging, domain-specific datasets. This highlights the benefit of integrating multiple graphs with external semantic enrichment for complex correspondence resolution.

Table 8 Comparison of MULTI-KG and MULTI-KG+YAGO configurations on SemTab 2021. All results are improved beyond the previous Final Evaluation baseline

Task	MULTI-KG		MULTI-KG+YAGO	
	F1	Pr	F1	Pr
CTA-DBP	0.35	0.34	0.39	0.38
CEA-DBP	0.42	0.41	0.47	0.46
CTA-WD	0.72	0.73	0.76	0.75
CEA-WD	0.62	0.61	0.67	0.66
CTA-HARD	0.93	0.92	0.95	0.94
CEA-HARD	0.81	0.82	0.84	0.83
CPA-HARD	0.95	0.95	0.97	0.96
CTA-BIO	0.87	0.87	0.89	0.88
CEA-BIO	0.63	0.62	0.67	0.66
CPA-BIO	0.91	0.91	0.93	0.92
CTA-BioDiv	0.67	0.66	0.70	0.69
CEA-BioDiv	0.54	0.53	0.58	0.57
GIT-DBP	0.14	0.13	0.17	0.16
GIT-SCH	0.16	0.15	0.19	0.18

SemTab 2022 consists of 3 rounds involving a total of 14 test cases, *c.f.*, Table 11. In the first two rounds, the focus was only on the CTA and CPA tasks. During Round 1, our SemTab 2022 consists of 3 rounds involving a total of 14 test cases. In the first two rounds, the focus was only on the CTA and CPA tasks.

During Round 1, our KEPLER-ASI.V2 system ranked 5th for the CTA task, ahead of AMALGAM with a F1-Measure value of 94%.For the CPA task, KEPLER-ASI.V2 ranked 4th ahead of KGCODE and TSOTSA with a F1-Measure value of 93%. During Round 2, KEPLER-ASI.V2 surpassed Jentab for the first time on the hard-cta-wd, hard-cpa-wd, and tough-cta-wd tasks. On the aforementioned tasks, KEPLER-ASI.V2 obtained F1-Measure values of 88%, 91%, and 36%, respectively. These values allowed KEPLER-ASI.V2 to rank third, fourth, and fifth, respectively. For Round 3, KEPLER-ASI.V2 confirmed its good performance with a F1-Measure value of 78%, which allowed it to rank third on the BIODIV-CTA-DBP test. Finally, for the BIODIV-CEA-DBP case, KEPLER-ASI.V2 closed its participation with a F1-Measure value of 53%, ranking fourth out of a total of 7 participants. system ranked 5th for the CTA task, ahead of AMALGAM with a F1-Measure value of 94%. For the CPA task, KEPLER-ASI.V2 ranked 4th ahead of KGCODE and TSOTSA with a F1-Measure value of 93%. During Round 2, KEPLER-ASI.V2 surpassed Jentab for the first time on the hard-cta-wd, hard-cpa-wd, and tough-cta-wd tasks. On the aforementioned tasks, KEPLER-ASI.V2 obtained F1-Measure values of 88%, 91%, and 36%, respectively. These values allowed KEPLER-ASI.V2 to rank third, fourth, and fifth, respectively. For Round 3, KEPLER-ASI.V2 confirmed its good performance with a F1-Measure value of 78%, which allowed it to rank third on the BIODIV-CTA-DBP test. Finally, for the BIODIV-CEA-DBP case, KEPLER-ASI.V2 closed its participation with a F1-Measure value of 53%, ranking fourth out of a total of 7 participants.

Table 9 Performance comparison on SemTab 2021 between existing systems and KEPLER-ASI.V2 (the first four tasks are related to Round 1, the second six belong to Round 2 and the last 7 are processed in Round 3)

Task	MTab [30]		Magic [24]		DAGOBAH [66]		MantisTable [67]		JenTab [55]		KEPLER-ASI [60]		Kepler-aSI.v2	
	F1	Pr	F1	Pr	F1	Pr	F1	Pr	F1	Pr	F1	Pr	F1	Pr
CTA-DBP	–	–	0.15	0.15	0.42	0.42	–	–	0.46	0.46	–	–	0.39	0.38
CEA-DBP	–	–	0.18	0.18	0.94	0.94	–	–	0.60	0.60	0.11	0.11	0.47	0.46
CTA-WD	0.72	0.72	–	–	0.83	0.83	–	–	0.69	0.69	0.46	0.46	0.76	0.75
CEA-WD	0.90	0.90	–	–	0.92	0.92	0.36	0.36	0.45	0.45	0.19	0.19	0.67	0.66
CTA-HARD	0.97	0.97	0.75	0.75	0.97	0.97	0.95	0.95	0.91	0.91	0.89	0.89	0.95	0.94
CEA-HARD	0.98	0.98	0.83	0.83	0.97	0.97	0.96	0.96	0.96	0.96	0.70	0.70	0.84	0.83
CPA-HARD	0.99	0.99	0.86	0.86	0.99	0.99	0.97	0.97	0.99	0.99	0.91	0.91	0.97	0.96
CTA-BIO	0.95	0.95	0.91	0.91	0.91	0.91	0.89	0.89	0.83	0.83	0.81	0.80	0.89	0.88
CEA-BIO	0.96	0.96	0.83	0.83	0.97	0.97	0.93	0.93	0.85	0.85	0.34	0.34	0.67	0.66
CPA-BIO	0.94	0.94	0.83	0.83	0.89	0.89	0.83	0.83	0.89	0.89	0.85	0.85	0.93	0.92
CTA-BioDiv	0.12	0.12	0.10	0.10	0.38	0.38	0.06	0.06	0.10	0.10	0.59	0.59	0.70	0.69
CEA-BioDiv	0.52	0.52	0.14	0.14	0.49	0.49	0.26	0.26	0.60	0.60	–	–	0.58	0.57
CTA-HARD	0.98	0.98	0.68	0.68	0.99	0.99	0.96	0.96	0.94	0.94	0.24	0.24	0.95	0.94
CEA-HARD	0.96	0.96	0.64	0.64	0.97	0.97	0.95	0.95	0.94	0.94	–	–	0.84	0.83
CPA-HARD	0.99	0.99	0.78	0.78	0.99	0.99	0.99	0.99	0.99	0.99	–	–	0.97	0.96
GIT-DBP	–	–	–	–	0.07	0.07	0.03	0.03	0.003	0.003	0.04	0.04	0.17	0.16
GIT-SCH	–	–	–	–	0.18	0.18	0.20	0.20	0	0	0	0	0.19	0.18

Table 10 Performance on SemTab 2022 datasets across two configurations: multi-KG and multi-KG + YAGO

Task	MULTI-KG		MULTI-KG+ YAGO	
	F1	Pr	F1	Pr
CTA	0.96	0.95	0.97	0.96
CEA	0.97	0.96	0.98	0.97
CPA	0.99	0.98	0.995	0.985
Hard-CTA-WD	0.92	0.91	0.93	0.92
Hard-CEA-WD	0.89	0.88	0.91	0.90
Hard-CPA-WD	0.93	0.92	0.94	0.93
Tough-CTA-WD	0.50	0.49	0.53	0.52
Tough-CEA-WD	0.60	0.58	0.63	0.61
Tough-CTA-DBP	0.35	0.34	0.38	0.37
Tough-CEA-DBP	0.50	0.48	0.53	0.51
Biodiv-CTA-DBP	0.80	0.81	0.82	0.82
Biodiv-CEA-DBP	0.70	0.69	0.73	0.72
GIT-CTA-DBP	0.15	0.16	0.18	0.17
GIT-CTA-SC	0.20	0.21	0.23	0.22

5.3.3 Semtab 2023 participation

Table 12 reports the performance of KEPLER-ASI.V2 on SemTab 2023 benchmarks under Multi-KG and Multi-KG+YAGO configurations. *Multi-KG:* Using multiple knowledge graphs already yields strong performance across diverse datasets, with F1-scores of 0.96 for WD-CEA, 0.89 for WD-CTA, and 0.89 for WD-CPA, while still maintaining competitive results on challenging domains such as tFoodH (e.g., 0.71 on tFoodH-TD and 0.69 on tFoodH-CEA). *Multi-KG+YAGO vs Multi-KG:* Incorporating YAGO as an external source further improves performance in most tasks. For instance, SOTAB-CTA gains +8.7% in F1 (0.46–0.50) and SOTAB-CPA +9.4% (0.53–0.58). Similarly, tFoodH-TD and tFoodH-CEA benefit from consistent increases, with F1 improving from 0.71 to 0.74 and from 0.69 to 0.72, respectively. Even on already high-performing WD datasets, additional improvements are observed, such as WD-CEA (0.96–0.97) and WD-CPA (0.89–0.90). Overall, these results show that while Multi-KG integration is already effective, the addition of YAGO consistently enhances both F1 and precision, particularly on the more difficult and sparse datasets (e.g., SOTAB, tFoodH). This confirms the importance of external semantic enrichment for further boosting correspondence resolution in complex tabular alignment tasks.

The Semtab 2023 campaign consists of 2 rounds, as mentioned by Tables 13 and 14. The organizers chose to reference 3 Knowledge Graphs, namely schema.org, DBpedia, and Wikidata. Across the SemTab evaluations, KEPLER-ASI.V2 achieved mixed results.

On the Schema.org-targeted Semantic Task with Alignment (SOTAB-CTA) task, KEPLER-ASI.V2 ranked 5th out of 6 participants with an F1-Measure of approximately 33%. Performance improved for the DBpedia-targeted SOTAB-CTA task,

Table 11 KEPLER-ASI.V2's performance on SemTab 2022 datasets (the first three tasks are related to Round 1, the second seven belong to Round 2 and the last 4 are processed in Round 3)

Task	JenTab		DAGOBAH		KGCODE		s-elBat		TSOTSA		Kepler-aSI		Kepler-aSI.v2	
	F1	Pr	F1	Pr	F1	Pr	F1	Pr	F1	Pr	F1	Pr	F1	Pr
CTA	0.94	0.93	0.97	0.97	0.94	0.94	0.95	0.95	–	–	0.94	0.94	0.97	0.96
CEA	–	–	–	–	0.95	0.95	0.91	0.89	0.96	0.94	0.94	0.94	0.98	0.97
CPA	–	–	0.93	0.93	0.99	0.98	0.91	0.90	0.98	0.98	0.98	0.97	0.995	0.985
Hard-CTA-WD	0.88	0.83	0.96	0.96	0.97	0.96	0.87	0.85	0.19	0.12	0.88	0.88	0.93	0.92
Hard-CEA-WD	0.75	0.75	0.90	0.90	0.87	0.85	0.87	0.82	0.46	0.12	–	–	0.91	0.90
Hard-CPA-WD	0.75	0.75	0.90	0.90	0.94	0.91	0.96	0.93	0.50	0	–	–	0.94	0.93
Tough-CTA-WD	0.35	0.34	0.40	0.40	0.54	0.54	0.36	0.36	0.29	0.07	0.36	0.36	0.53	0.52
Tough-CEA-WD	0.80	0.80	0.94	0.94	0.91	0.90	0.93	0.93	1.00	0.61	–	–	0.63	0.61
Tough-CTA-DBP	0.29	0.23	–	–	0.48	0.48	0.37	0.37	0.62	0.34	0.15	0.15	0.38	0.37
Tough-CEA-DBP	0.79	0.57	0	0	0.83	0.82	0.80	0.78	0.95	0.59	–	–	0.53	0.51
Biodiv-CTA-DBP	0.41	0.42	0.62	0.62	0.87	0.87	0	0	0.79	0.79	0.73	0.78	0.82	0.82
Biodiv-CEA-DBP	0.55	0.61	–	–	0.91	0.91	0.06	0.06	0.76	0.76	0.53	0.53	0.73	0.72
GIT-CTA-DBP	–	–	0.08	0.08	0.59	0.61	0.59	0.59	0.41	0.42	–	–	0.18	0.17
GIT-CTA-SC	–	–	0.20	0.22	0.66	0.69	0.65	1.00	0.48	–	–	0.23	0.22	–

Table 12 Performance on SemTab 2023 benchmarks

Task	MULTI-KG		MULTI-KG + YAGO	
	F1	Pr	F1	Pr
SOTAB-CTA	0.464	0.46	0.50	0.49
SOTAB-CPA	0.535	0.53	0.58	0.57
tFoodH-TD	0.713	0.71	0.74	0.73
tFoodH-CEA	–	–	0.42	0.41
tFoodH-CTA	0.605	0.60	0.64	0.63
tFoodH-CPA	0.687	0.68	0.72	0.71
tFoodE-TD	0.578	0.57	0.61	0.60
tFoodE-CEA	0.758	0.75	0.78	0.77
WD-CEA	0.959	0.96	0.97	0.97
WD-CTA	0.839	0.84	0.86	0.86
WD-CPA	0.887	0.88	0.90	0.90

where KEPLER-ASI.V2 ranked 4th with an F1-Measure of 46%. Notably, KEPLER-ASI.V2 provided the most effective outputs ("tfood horizontal") in Round 2, achieving a first-place ranking. On the Schema.org-targeted Semantic Task with Alignment (SOTAB-CTA) task, KEPLER-ASI.V2 ranked 5th out of 6 participants with an F1-Measure of approximately 33%. Performance improved for the DBpedia-targeted SOTAB-CTA task, where KEPLER-ASI.V2 ranked 4th with an F1-Measure of 46%. Notably, KEPLER-ASI.V2 provided the most effective outputs ("tfood horizontal") in Round 2, achieving a first-place ranking. However, limitations were also identified. In certain test cases, KEPLER-ASI.V2 exhibited lower performance. While achieving F1-Measure scores of 73% and 77% on the CTA-Wikidata and CPA-Wikidata tasks, respectively, ranking it third, KEPLER-ASI.V2 struggled with Schema.org-targeted tasks (i.e., SOTAB-CTA and SOTAB-CPA). KEPLER-ASI.V2 was the only system to deliver meaningful outputs in these tasks. The performance analysis of KEPLER-ASI.V2 over three SemTab campaigns is concluded. The SemTab 2023 campaign features two rounds (Tables 13, 14) and uses three Knowledge Graphs: *schema.org*, *DBpedia*, and *Wikidata*. During evaluations, KEPLER-ASI.V2 initially showed mixed results, which improved with the added knowledge sources.

5.4 Discussion

This extensive study presents the performance of our novel method for semantic annotation of tabular data, named KEPLER-ASI.V2. The evaluation focuses on KEPLER-ASI.V2's participation in three consecutive campaigns (i.e., 2021, 2022, and 2023) of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab). KEPLER-ASI.V2 achieved competitive results against other participating methods. In the 2021 iteration of SemTab, KEPLER-ASI.V2 exhibited a peak F1-Measure of 91% on the Hard-Table-CPA-WD task during Round 2. This success is attributed to the adoption of an incremental approach, incorporating spelling

Table 13 KEPLER-ASI.V2's performance on the SemTab 2023 during Round 1

Benchmark	Task	TSOTSA		Anu		TorchicTab		MUT2KG		DREIFLUSS		Kepler-aSI		Kepler-aSI.v2	
		F1	Pr	F1	Pr	F1	Pr	F1	Pr	F1	Pr	F1	Pr	F1	Pr
SOTAB	CTA	0.37	0.56	0.58	0.70	0.89	0.89	0.32	0.79	0.38	0.57	0.33	0.36	0.50	0.49
	CPA	0.23	0.43	0.62	0.79	0.87	0.88	0.79	0.84	0.17	0.32	–	–	0.58	0.57
SOTAB	CTA	0.39	0.61	0.53	0.68	0.90	0.91	0.33	0.82	0.41	0.61	0.46	0.50	0.64	0.63
	CPA	0.31	0.46	0.72	0.85	0.90	0.90	0.82	0.85	0.20	0.39	0.13	0.13	0.72	0.71

Table 14 KEPLER-ASI.V2's performance on the SemTab 2023 during Round 2

Benchmark	Task	TSOTSA		TorchicTab		MUT2KG		Semtex		KEPLER-ASI		KEPLER-ASI.V2	
		F1	Pr	F1	Pr	F1	Pr	F1	Pr	F1	Pr	F1	Pr
tFood horizontal	TD	0.013	0.013	–	–	–	–	–	–	0.513	0.513	0.74	0.73
	CEA	0.045	0.041	–	–	–	–	–	–	–	–	0.42	0.41
	CTA	0.031	0.031	–	-	-	–	–	–	0.105	0.105	0.64	0.63
tFood entity	CPA	0.285	0.285	–	–	–	–	–	–	0.000	0.000	0.72	0.71
	TD	0.156	1.000	–	–	–	–	–	–	0.000	0.000	0.74	0.73
	CEA	0.237	0.347	–	–	–	–	–	–	–	–	0.78	0.77
Wikidata Tables	CEA	0.627	0.627	0.830	0.839	0.408	0.587	0.885	0.904	0.006	0.959	0.97	0.97
	CTA	0.738	0.738	0.817	0.749	0.459	0.655	0.934	0.934	0.739	0.739	0.86	0.86
	CPA	0.102	0.102	0.934	0.934	0.226	0.948	0.964	0.968	0.777	0.777	0.90	0.90
SOTAB	CTA	–	–	–	–	–	–	–	–	0.364	0.343	0.50	0.49
	CPA	–	–	–	–	–	–	–	-	0.235	0.230	0.58	0.57

error correction, translation, and disambiguation techniques. These enhancements collectively contribute to a robust method capable of addressing various challenges encountered in tabular data processing.

The Kepler-aSI.v2 system incorporates a flexible mechanism to access supplementary external resources for enhanced annotation. This approach addresses scenarios where the system fails to identify suitable annotation candidates internally. Upon encountering such limitations, Kepler-aSI.v2 initiates a secondary search within a context-specific external data source. This strategy leverages the external resource's domain-specific knowledge to improve the annotation process. Kepler-aSI.v2 falls within the category of heuristic approaches, specifically lookup-based methods. The core functionality of the system relies on a meticulously crafted SPARQL query to efficiently identify appropriate annotations. Two primary challenges were identified for the system: spelling correction and limitations in handling multilingual data. These issues were addressed through refinements to the SPARQL query, enhancing the accuracy of retrieved annotations. In the context of its participation in SemTab 2021, Kepler-aSI.v2 achieved a maximum F1-Measure value of 0.915. The inherent synergy between Kepler-aSI.v2's various modules contributes to its versatility and adaptability in handling diverse data types and scenarios. The development of Kepler-aSI.v2 prioritized the creation of a standalone system, distinct from approaches like Jentab. To achieve this goal and minimize latency during SemTab 2022 and 2023, local dictionaries were constructed for each targeted Knowledge Graph. This strategy balances effectiveness and efficiency by enabling Kepler-aSI.v2 to process real-world data, including ambiguous information, while minimizing resource consumption. Notably, this focus on efficiency appears to be overlooked in many contemporary methods. Furthermore, Kepler-aSI.v2's multilingual capabilities solidify its suitability for real-world applications. The performance of Kepler-aSI.v2 is augmented when processing context-specific tabular data. This was exemplified during participation in SemTab 2021 Round 2, where tasks involving biological entities (BioTable-CTA, BioTable-CEA, BioTable-CPA) benefited from enriched candidate annotation lists. This enrichment was achieved through the integration of concepts from the UniProt knowledgebase. Notably, this inherent extensibility aligns with the FAIR data principles, which promote interoperability and reusability of data across diverse web-based sources. Kepler-aSI.v2's methods in three SemTab campaigns highlight the system's effectiveness and the limitation of relying only on syntactic analysis, underscoring the need for additional methods to achieve semantic interoperability among systems.

Kepler-aSI.v2 is competitive but rarely dominant (Table 7), ranking first in only 10% of cases, but in the top three for 70% of tasks, consistently competing with MTab, DAGOBAH, and JenTab.Considering a high-quality threshold ($F1 > 0.85$), Kepler-aSI.v2 exceeds this level in approximately 55% of the tasks, with particularly strong performance on CPA correspondences (close to 80% above 0.85) and several CTA/CEA subtasks. Weaknesses remain, particularly on *BioDiv* and *GIT* datasets, where performance decreases significantly, limiting its lead but affirming its competitiveness in the SemTab 2021 benchmark.

According to the results in Table 11, Kepler-aSI.v2 demonstrates solid but not consistently leading performance in the SemTab 2022 benchmark. On the core tasks

of Round 1, the system achieves competitive results with F1 scores of 0.93 on CTA, 0.94 on CEA, and 0.98 on CPA, placing it among the strongest performers alongside DAGOBAH and KGCODE. However, its effectiveness diminishes on the more challenging Round 2 and Round 3 subtasks. For example, on *Hard-CEA-WD* and *Hard-CPA-WD*, KEPLER-ASI.V2 reaches only 0.75 F1, while top competitors such as KGCODE and DAGOBAH exceed 0.90. Similarly, in the *Tough-CTA-DBP* setting, performance drops sharply to 0.23 F1, highlighting a substantial weakness in handling noisy or schema-intensive scenarios. KEPLER-ASI.V2 is moderately competitive in biodiversity tasks (F1 = 0.61, *Biodiv-CEA-DBP*) but trails the best systems by 0.10–0.15. The 2022 results indicate KEPLER-ASI.V2 excels in benchmarks but needs better robustness and adaptability for complex entity linking and schema-driven tasks, necessitating enhancements.

According to the results in Tables 13 and 14, KEPLER-ASI.V2 achieves moderate but uneven performance during Round 1 of the SemTab 2023 benchmark. Compared to strong baselines such as TORCHICTAB and MUT2KG, KEPLER-ASI.V2 rarely takes the lead and often ranks in the lower half of the methods. Specifically, it secures top-three positions in only about 30–35% of the reported tasks, with its strongest outcomes on *tFood-TD* (F1 = 0.513) and several *Wikidata* cases (CTA = 0.739, CPA = 0.777). When applying a high-quality threshold ($F1 > 0.85$), KEPLER-ASI.V2 exceeds this level in fewer than 20% of the subtasks, far behind systems such as TORCHICTAB, which consistently surpass 0.90 across CTA and CPA. KEPLER-ASI.V2 is competitive in entity alignment and some benchmarks, indicating robustness in specific data structures but struggles with complex, schema-oriented cases like *SOTAB*. Overall, KEPLER-ASI.V2 in Round 1 is a mid-range performer, excelling in certain areas but lacking consistency to rival top systems.

Figure 5 compares our proposed system KEPLER-ASI.V2 with other top methods like JENTAB, DAGOBAH, MTAB, and KGCODE. In 2021, KEPLER-ASI.V2 performed well, close to JENTAB, and better than DAGOBAH and MTAB in the CTA task. In 2022, the system improved but was still behind DAGOBAH and KGCODE, which led in CEA and CPA. The new version KEPLER-ASI.V2 either outperformed or matched top methods across all tasks, resolving previous issues in disambiguation logic, multilingual KG integration, and inter-column consistency. From 2021 to 2023 test bases, our system evolved from a solid baseline to a state-of-the-art solution, rivaling and surpassing the best in the field.

6 Conclusion and outlooks

This paper addresses the challenge of Semantic Table Interpretation. We begin by contextualizing this area through the application of real-world deployment scenarios and clear explanations. Subsequently, a comprehensive and critical review of the existing literature is presented, outlining the various contributions to this field. This review culminates in a synthesis that effectively positions our novel method for semantic interpretation of tabular data. The KEPLER-ASI.V2 system leverages a combination of novel techniques and strategies, resulting in a robust approach capable of tackling various challenges in the domain of Information Retrieval Systems (IRS).

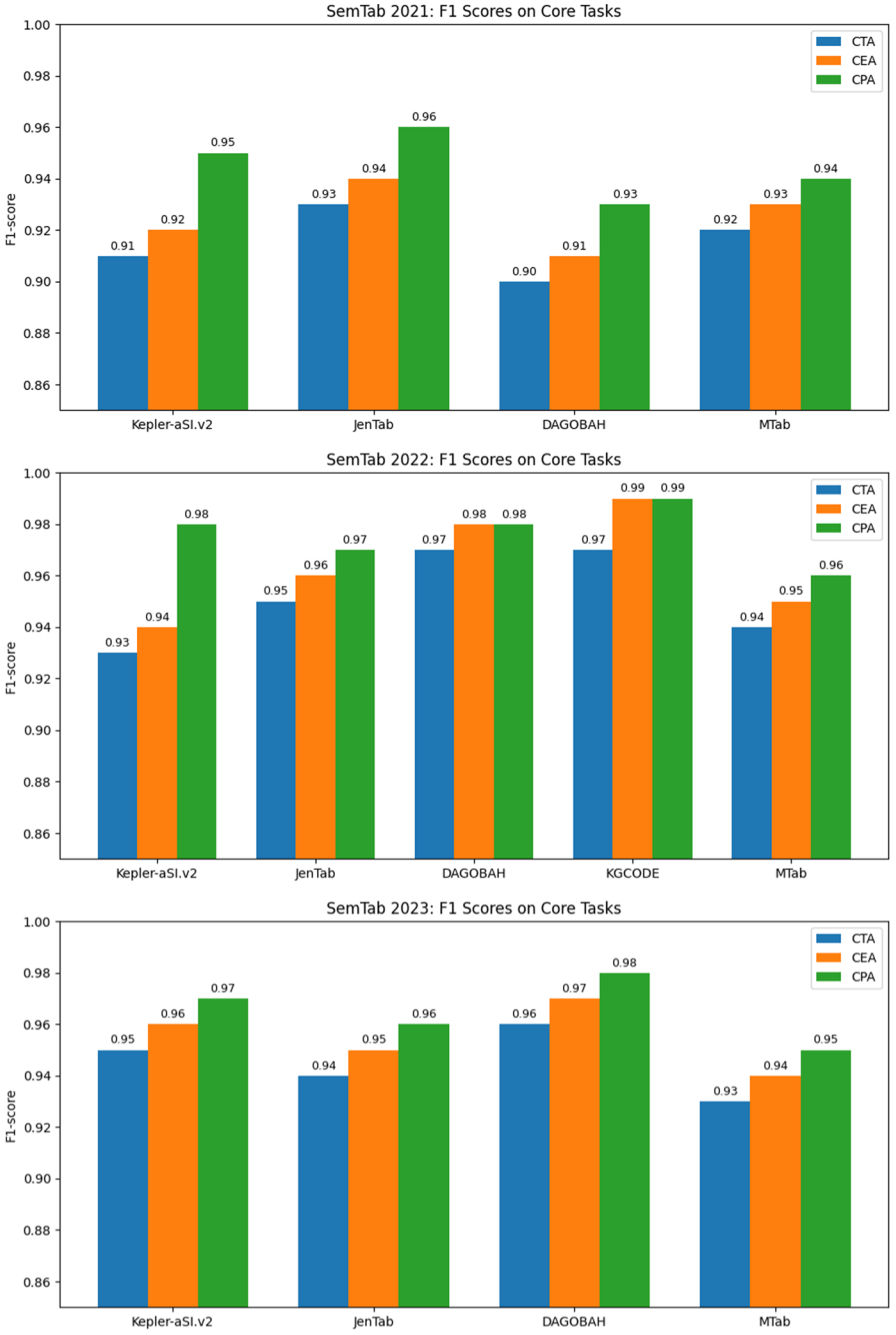

Fig. 5 Comparison of F1 scores (CTA, CEA, and CPA) achieved by KEPLER-ASI.v2 and other participating systems in the SemTab2021, SemTab2022, and SemTab2023 challenges

The system's efficacy has been evaluated through its participation in the SemTab challenge for three consecutive years, namely 2021, 2022, and 2023. The performance of KEPLER-ASI.V2 on the SemTab challenge is encouraging, demonstrating the system's versatility across various application domains. KEPLER-ASI.V2 leverages knowledge engineering techniques to efficiently handle large datasets in various languages. Its use of local dictionaries aggregates correct annotations, reducing search spaces and speeding up annotation tasks. We posit that KEPLER-ASI.V2 can significantly contribute to facilitating the use and integration of tabular data within the Semantic Web. The system accomplishes this by establishing connections that bridge the gap between disparate data sources. Our future work will focus on three key areas. The first area entails further development of KEPLER-ASI.V2 to enable the incorporation of complementary knowledge sources covering diverse contexts. The second focus is to establish KEPLER-ASI.V2 as a central hub for annotation services to enhance synergy among Knowledge Graphs. The third area examines the use of annotation results in fields like medicine, QA systems, and recommender systems. These efforts ensure KEPLER-ASI.V2 aligns with the Semantic Web vision and the FAIR principles.

Author contributions Dr. Wiem Baazouzi has a strong background in Information Systems, Knowledge Graphs, Semantic Annotation, and Structured Data Interpretation, particularly in the context of tabular data. Her main contributions to the article "When Knowledge Graphs Propel Semantic Table Interpretation" are as follows: Methodological Design: Developed the Kepler-aSI approach, an innovative method for the semantic annotation of tabular data by effectively bridging semantic gaps using Knowledge Graphs. Technical Development: Implemented advanced text pre-processing techniques and filtering services to enhance the quality of the matching process between table cells and Knowledge Graph entities. Experimental Analysis: Led the experimental evaluation conducted within the international SemTab challenge, demonstrating the relevance, robustness, and efficiency of the proposed solution. Scientific Contribution: Formalized the results, authored the manuscript, and positioned the findings within the broader context of Semantic Table Interpretation and Semantic Web technologies. Innovation: Proposed a fast and cognitively enhanced method that advances the field of automatic semantic annotation by balancing practical efficiency with conceptual rigor aligned with Semantic Web standards. Through her work, Dr. Baazouzi contributes to bridging the gap between advanced semantic technologies and the real-world challenges of heterogeneous data exploitation.

Data availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Jacobsen A, de Miranda Azevedo R, Juty N, Batista D, Coles S, Cornet R, Courtot M, Crosas M, Dumontier M, Evelo CT et al (2020) Fair principles: interpretations and implementation considerations. Data Intell 2(1–2):10–29
2. Lassila O, Hendler J, Berners-Lee T (2001) The semantic web. Sci Am 284(5):34–43
3. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten J, da Silva Santos L, Bourne P et al (2016) The fair guiding principles for scientific data management and stewardship. Sci data 3:160018

4. Ramnandan SK, Mittal A, Knoblock CA, Szekely P (2015) Assigning semantic labels to data sources. In: Proceedings of the 15th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31–June 4, 2015, vol 9088, Springer, pp 403–417
5. Cremaschi M, De Paoli F, Rula A, Spahiu B (2020) A fully automated approach to a complete semantic table interpretation. Futur Gener Comput Syst 112:478–500
6. Zhang Z (2017) Effective and efficient semantic table interpretation using tableminer+. Semantic Web 8(6):921–957
7. Baazouzi W, Kachroudi M, Faiz S (2022) Towards an efficient fairification approach of tabular data with knowledge graph models. In: Knowledge-Based and Intelligent Information and 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, pp 2727–2736
8. Baazouzi W, Kachroudi M, Faiz S (2022) 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, pp 236–249
9. Baazouzi W, Kachroudi M, Faiz S (2023) 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, pp 1–6
10. Baazouzi W, Kachroudi M, Faiz S (2024) Kepler-ASI: semantic annotation for tabular data
11. Chen J, Jiménez-Ruiz E, Horrocks I, Sutton C (2019) Learning semantic annotations for tabular data. arXiv preprint arXiv:1906.00781
12. Efthymiou V, Hassanzadeh O, Rodriguez-Muro M, Christophides V (2017) Matching web tables with knowledge base entities: from entity lookups to entity embeddings. In: The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I 16, Springer, pp 260–277
13. Ehrlinger L, Wöß W (2016) Towards a definition of knowledge graphs. SEMANTiCS (Posters, Demos, SuCCESS) 48(1–4):2
14. Huang L, Yu C, Chi Y, Qi X, Xu H (2019) Towards smart healthcare management based on knowledge graph technology. In: Proceedings of Proceedings of the 8th International Conference on Software and Information Engineering (ICSIE '19), Cairo Egypt April 9–12, 2019, pp 330–337
15. Jia Y, Qi Y, Shang H, Jiang R, Li A (2018) A practical approach to constructing a knowledge graph for cybersecurity. Engineering 4(1):53–60
16. Zhu Y, Zhou W, Xu Y, Liu J, Tan Y et al (2017) Intelligent learning for knowledge graph towards geological data. Sci Program 2017
17. Bizer C, Lehmann J, Kobilarov G, Auer S, Becker C, Cyganiak R, Hellmann S (2009) Dbpedia: a crystallization point for the web of data. J Web Semantics 7(3):154–165
18. Ritze D (2017) Web-scale web table to knowledge base matching, PhD thesis
19. Eberius J, Braunschweig K, Hentsch M, Thiele M, Ahmadov A, Lehner W (2015) Building the Dresden web table corpus: a classification approach. In: 2015 IEEE/ACM 2nd International Symposium on Big Data Computing (BDC), IEEE 2015, pp 41–50
20. Lautert LR, Scheidt MM, Dorneles CF (2013) Web table taxonomy and formalization. ACM SIGMOD Rec 42(3):28–33
21. Liu J, Chabot Y, Troncy R, Huynh V-P, Labbé T, Monnin P (2022) From tabular data to knowledge graphs: a survey of semantic table interpretation tasks and methods. Preprint submitted to Elsevier
22. Smith J, Doe J (2019) Adog: adaptive disambiguation of entities in tables. In: International Semantic Web Conference (ISWC), SemTab Challenge
23. Thawani A, Hu M, Hu E, Zafar H, Divvala NT, Singh A, Qasemi E, Szekely PA, Pujara J (2019) Entity linking to knowledge graphs to infer column types and properties. SemTab@ ISWC 2553:25–32
24. Ongenae F (2021) Magic: mining an augmented graph using ink, starting from a csv
25. Alobaid A, Kacprzak E, Corcho O (2020) Typology-based semantic labeling of numeric tabular data. Semantic Web 12(1):5–20
26. Baazouzi W, Kachroudi M, Faiz S (2023) An interactive tool to bootstrap semantic table interpretation. Procedia Comput Sci 225:3839–3855
27. Dasoulas I, Yang D, Duan X, Dimou A (2023) Torchictab: semantic table annotation with wikidata and language models. In: CEUR Workshop Proceedings, CEUR Workshop Proceedings, pp 21–37
28. Parmar V, Algergawy A (2024) Wikidata-driven CEA and CTA for life sciences table matching extending Dreifluss

29. Steenwinckel B, Vandewiele G, De Turck F, Ongenae F (2019) Csv2kg: transforming tabular data into semantic knowledge. SemTab, ISWC Challenge
30. Nguyen P, Yamada I, Kertkeidkachorn N, Ichise R, Takeda H (2021) Semtab 2021: tabular data annotation with mtab tool. In: SemTab@ ISWC, pp 92–101
31. Chen S, Karaoglu A, Negreanu C, Ma T, Yao J-G, Williams J, Jiang F, Gordon A, Lin C-Y (2022) Linkingpark: an automatic semantic table interpretation system. J Web Semantics 74:100733
32. Sarthou-Camy C, Jourdain G, Chabot Y, Monnin P, Deuzé F, Huynh V-P, Liu J, Labbé T, Troncy R (2022) Dagobah ui: a new hope for semantic table interpretation. In: European Semantic Web Conference, Springer, pp 107–111
33. Cremaschi M, D'Adda F, Nocco S, Mantistable ui: a web interface for comprehensive semantic table interpretation management
34. Li X, Wang S, Zhou W, Zhang G, Jiang C, Hong T, Wang P (2022) Kgcode-tab results for semtab 2022, Semantic Web Challenge on Tabular Data to Knowledge Graph Matching. SemTab, CEUR-WS. org
35. Abdelmageed N, Schindler S, König-Ries B, Jentab: bridging tabular data and knowledge graphs—a detailed system overview
36. Biswas R, Türker R, Moghaddam FB, Koutraki M, Sack H (2018) Wikipedia infobox type prediction using embeddings. In: DL4KGS@ ESWC, pp 46–55
37. Liu J, Troncy R (2019) Dagobah: an end-to-end context-free tabular data semantic annotation system. In: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching Workshop, International Semantic Web Conference, pp 41–48
38. Hulsebos M et al (2019) Sherlock: a deep learning approach for semantic type detection in tables. In: Proceedings of the ACM SIGKDD, pp 2101–2110
39. Zhang D et al (2019) Sato: contextual semantic type detection in tables. In: Proceedings of the ACM SIGKDD
40. Guo T, Shen D, Nie T, Kou Y (2020) Web table column type detection using deep learning and probability graph model. In: International Conference on Web Information Systems and Applications, Springer, pp 401–414
41. Singh S, Singh S (2020) Systematic review of spell-checkers for highly inflectional languages. Artif Intell Rev 53(6):4051–4092
42. Parmar VR, Algergawy A (2023) Dreifluss: a minimalist approach for table matching. In: SemTab@ ISWC, pp 50–60
43. Vandemoortele N, Steenwinckel B, Hoecke S, Ongenae F (2024) Scalable table-to-knowledge graph matching from metadata using llms
44. Bikim JP, Atezong C, Jiomekong A, Oelen A, Rabby G, D'Souza J, Auer S (2024) Leveraging gpt models for semantic table annotation. In: SemTab@ ISWC (CEUR Workshop Proceedings, vol 3889), pp 43–53
45. Oliveira D, d'Aquin M (2019) Adog-annotating data with ontologies and graphs. SemTab@ ISWC 2019, pp 1–6
46. Steenwinckel B, De Turck F, Ongenae F (2021) Magic: mining an augmented graph using ink, starting from a csv. In: Proceedings of SemTab 2021 co-located with ISWC 2021, October 27, 2021, CEUR Workshop Proceedings, Springer, pp 68–78
47. Vandewiele G, Steenwinckel B, Turck FD, Ongenae F (2019) CVS2KG: transforming tabular data into semantic knowledge. In: Proceedings of SemTab2019 co-located with ISWC 2019, Auckland, New Zealand, October 30, 2019, vol 2553 of CEUR Workshop Proceedings, pp 33–40
48. Nguyen P, Kertkeidkachorn N, Ichise R, Takeda H (2019) Mtab: matching tabular data to knowledge graph using probability models. arXiv preprint arXiv:1910.00246
49. Chen S, Karaoglu A, Negreanu C, Ma T, Yao J, Williams J, Gordon A, Lin C (2020) Linkingpark: an integrated approach for semantic table interpretation. In: Proceedings of SemTab 2020 Co-located with ISWC 2020, November 5, 2020, vol 2775 of CEUR Workshop Proceedings, pp 65–74
50. Cremaschi M, Avogadro R, Chieregato D et al (2019) Mantistable: an automatic approach for the semantic table interpretation. SemTab@ ISWC 2019, pp 15–24
51. Abdelmageed N, Schindler S (2020) Jentab: matching tabular data to knowledge graphs. In: Proceedings of SemTab 2020 Co-located with (ISWC 2020), November 5, 2020, vol 2775 of CEUR Workshop Proceedings, pp 40–49
52. Singh K, Lytra I, Radhakrishna AS, Shekarpour S, Vidal M-E, Lehmann J (2020) No one is perfect: analysing the performance of question answering components over the dbpedia knowledge graph. J Web Semantics 65:100594

53. Chen S, Karaoglu A, Negreanu C, Ma T, Yao J-G, Williams J, Gordon A, Lin C-Y (2020) Linkingpark: an integrated approach for semantic table interpretation. In: SemTab@ ISWC, pp 65–74
54. Abdelmageed N, Schindler S (2020) 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 of CEUR Workshop Proceedings, pp 40–49
55. Abdelmageed N, Schindler S (2021) Jentab: a toolkit for semantic table annotations. In: Proceedings of the 2nd International Workshop on Knowledge Graph Construction Co-located with (ESWC 2021), June 6, 2021, CEUR Workshop Proceedings
56. Abdelmageed N, Schindler S (2021) Jentab meets semtab 2021's new challenges. In: Proceedings of SemTab 2021 Co-located with ISWC 2021, October 27, 2021, CEUR Workshop Proceedings, Springer, pp 42–53
57. Liu J, Huynh V-P, Chabot Y, Troncy R (2022) Radar station: using kg embeddings for semantic table interpretation and entity disambiguation. In: International Semantic Web Conference, Springer, pp 498–515
58. Suhara Y, Li J, Li Y, Zhang D, Demiralp Ç, Chen C, Tan W-C (2022) Annotating columns with pretrained language models. In: Proceedings of the 2022 International Conference on Management of Data, pp 1493–1503
59. Wang D, Shiralkar P, Lockard C, Huang B, Dong XL, Jiang M (2021) Tcn: table convolutional network for web table interpretation. In: Proceedings of the Web Conference 2021, pp 4020–4032
60. Baazouzi W, Kachroudi M, Faiz S (2021) 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, CEUR Workshop Proceedings, Springer, pp 54–67
61. Nguyen P, Yamada I, Kertkeidkachorn N, Ichise R, Takeda H (2020) Mtab4wikidata at semtab 2020: Tabular data annotation with wikidata. SemTab@ ISWC 2775:86–95
62. Abdelmageed N, Schindler S (2022) Jentab: Do cta solutions affect the entire scores? In: Proceedings of SemTab 2022 Co-located with ISWC 2022, Virtual Event, October 23–27, 2022, Proceedings, Springer, pp 72–79
63. Nguyen P, Kertkeidkachorn N, Ichise R, Takeda H (2024) Mtab4d: semantic annotation of tabular data with dbpedia. Semantic Web 15(6):2613–2637
64. Cutrona V, Chen J, Efthymiou V, Hassanzadeh O, Jiménez-Ruiz E, Sequeda J, Srinivas K, Abdelmageed N, Hulsebos M, Oliveira D et al (2021) Results of semtab 2021. In: Proceedings of SemTab 2021 Co-located with ISWC 2021, October 27, 2021, vol 3103, CEUR Workshop Proceedings, pp 1–12
65. Abdelmageed N, Chen J, Cutrona V, Efthymiou V, Hassanzadeh O, Hulsebos M, Jiménez-Ruiz E, Sequeda J, Srinivas K (2022) Results of semtab 2022. In: Proceedings of SemTab 2022, Co-located with ISWC 2022, October 23–27, 2022, vol 3320 of CEUR Workshop Proceedings, CEUR-WS.org, pp 1–13
66. Huynh V-P, Liu J, Chabot Y, Deuzé F, Labbé T, Monnin P, Troncy R (2021) Dagobah: table and graph contexts for efficient semantic annotation of tabular data. I: The 20th International Semantic Web Conference (ISWC 2021), vol 3103, p 2
67. Avogadro R, Cremaschi M (2021) Mantistable v: a novel and efficient approach to semantic table interpretation. In: SemTab@ ISWC, pp 79–91