An Explainable Header-Centric Framework for Large-Scale Semantic Table Interpretation and Data Quality Assessment

Marcelo Valentim Silva[1,*], Hannes Herrmann[2] and Valerie Maxville[3]

Curtin University, Kent Street, Bentley, Perth, Western Australia, 6102, Australia

Abstract

Knowledge Graph (KG) quality is influenced not only by downstream graph validation, but also by the quality of the tabular metadata used prior to integration. In metadata-only Semantic Table Interpretation (STI), where cell values are unavailable, noisy, or unsuitable for use, column headers become a critical source of semantic evidence. This is directly relevant to metadata quality, provenance-aware traceability, and quality-aware preparation for knowledge-graph integration.

We present an explainable, header-centric framework for metadata-only Column Type Annotation (CTA) and Data Quality Assessment (DQA). The framework maps headers to a compact layer of 39 interpretable *FinalFormat* types using curated lexical resources, while preserving token-level traceability through *SourceKeywords*. Each assigned type activates predefined validation rules grounded in a structured taxonomy of Data Quality Issues (DQI), producing cell-level DQI instances such as missing data, duplicates, domain violations, wrong data type, and temporal mismatch. These detections are aggregated into *HeadersIQ*, a lightweight and intentionally unweighted data source-level data quality metric relevant to the SemTab 2024 IsGold? track and related KG quality assessment work, such as KG2Tables. Illustrative case studies from real data sources show that identifier semantics simultaneously trigger completeness and uniqueness constraints, whereas bounded numerical formats expose domain violations.

The framework was evaluated across heterogeneous benchmarks, including UCI, Prague, Kaggle, VizNet/Sato, SOTAB, T2Dv2, and the SemTab 2024 Metadata-to-KG track, comprising around 120,000 header columns. This large-scale evaluation demonstrates broad practical coverage across noisy real-world metadata, while a parallel KG-mapping pathway supports alignment to DBpedia and Schema.org. On the SemTab 2024 Metadata-to-KG track, the official GT-strict evaluation was modest. Still, a blinded diagnostic audit indicates that many mismatches reflect benchmark granularity, aliasing, and ontology-selection effects rather than wholly implausible header-centric predictions. We report this audit as diagnostic evidence on the structure of disagreements, not as revised benchmark performance. Overall, the paper presents a reusable and auditable workflow for metadata-driven semantic annotation, data source-level quality monitoring, and KG-oriented benchmark diagnosis. This positions the framework as a practical contribution to quality-aware tabular preparation and the broader goal of sustaining trustworthy knowledge graph ecosystems.

Keywords

Semantic table interpretation, column type annotation, knowledge graphs, data quality assessment

1. Introduction

In recent years, the rapid expansion of open data repositories and domain-specific tabular data sources has made Table Understanding (TU) a central challenge for semantic integration and interoperability with knowledge graphs [1]. Most real-world tables remain poorly annotated or lack explicit semantic types, making semantic integration difficult and prone to errors. Within TU, Semantic Table Interpretation (STI) addresses this need by linking tabular data to ontologies and knowledge graphs such as DBpedia [2] and Schema.org [3]. In this setting, the Column Type Annotation (CTA) subtask assigns semantically meaningful types to table columns, supporting semantic alignment and enabling richer metadata for downstream data quality assessment and knowledge-graph integration.

Workshop on Quality of Knowledge Graphs at ESWC 2026, May 11, 2026, Dubrovnik, Croatia

*Corresponding author.

marcelo.valentim@bcb.gov.br (M. V. Silva); hannes.herrmann@curtin.edu.au (H. Herrmann); v.maxville@curtin.edu.au (V. Maxville)

0000-0003-4592-2968 (M. V. Silva); 0000-0002-3160-4988 (H. Herrmann); 0000-0001-8842-7364 (V. Maxville)

Student **ID**	Last **Name**	First**Name**	**Age**	**Country**	**Height**	Birth**Date**
I345343	white	3	200	USA	145	3/04/2121
J849486	Stewart	Ronald	28	?	170	0/1/2010
J849486	Johnson	Mary	56	Australia	-200	NULL

Figure 1: Example of Bad Data/Dirty Dataset.

Figure 1 illustrates a realistic "dirty" dataset with duplicates, missing values, and obvious typographical problems. In our framework, such problems are detected and explained through the semantic types inferred from the terms that appear in bold in the column headers.

1.1. Historical context and renewed interest

Influential research such as **Sherlock (2019) [4]** discouraged header-centric features, arguing that matching-based approaches (i) recognise only a limited range of types and (ii) are not robust to malformed or context-dependent data, particularly when relying on dictionary or regular-expression matching. This influenced community practice, where unrecognised headers often fall back to primitive types.

Recent findings challenge this view. A survey by **Liu et al. (2023) [5]** reports that competitive CTA systems increasingly exploit header signals: ***"The table header often directly explains the contents of the column"*** and that using header information can help predict column types or properties more efficiently.

Since 2019, the Semantic Web Challenges on Tabular Data to Knowledge Graph Matching (SemTab Challenges) [6, 7, 8, 9, 10, 1] have reinforced this shift. In particular, the SemTab 2024 edition [9] introduced a Metadata-to-KG track in which cell values could not be used, reflecting practical concerns such as privacy, sensitive data, and large-scale processing. SemTab 2024 also included the **IsGold? track [11]**, which further emphasized automated **data quality assessment**. **KG2Tables (Abdelmageed et al., 2025) [12]** also highlighted data quality as an open gap (diversity, annotation accuracy/clarity, structural coherence). These developments further reinforce the relevance of scalable and interpretable metadata-based methods.

In particular, AdaTyper (2023) [13], from the Sherlock/Sato group, prioritises header analysis before cell-based fallback, further supporting the value of header semantics.

Taken together, these developments motivate an approach that mitigates the limitations noted by Sherlock [4]. Dynamic dictionary expansion increases coverage beyond the limited type range typically associated with matching-based systems. Normalisation and abbreviation mappings improve robustness to noisy, malformed, and lexically variable metadata that would otherwise defeat dictionary-based matching. In addition, the ***FinalFormat*** layer extends semantic coverage beyond simple primitive categories, although context-dependent ambiguity remains a challenging case.

1.2. Approach and Contributions

This work extends a previous header-centric semantic typing and data quality framework [14] into a broader framework for metadata-only Semantic Table Interpretation (STI) and Data Quality Assessment (DQA). Its central premise is that column headers can provide the primary semantic evidence needed for interpretable column typing and downstream quality assessment, especially when cell values are noisy, unavailable, sensitive, or unsuitable for use.

The framework follows a header-centric workflow in which headers are normalised and expanded through curated lexical resources. After preprocessing, the pipeline branches into two parallel pathways. In the first, semantic type detection maps headers to interpretable semantic types in the ***FinalFormat*** layer. These assignments activate targeted data quality checks grounded in a structured taxonomy of **Data Quality Issues**. In the second, a separate KG-mapping pathway aligns preprocessed headers to DBpedia and Schema.org for KG-oriented interpretation and benchmark comparison.

Traceability is preserved through ***SourceKeywords***, and detected violations can be aggregated through ***HeadersIQ*** for data source-level quality monitoring. The framework is then evaluated across heterogeneous benchmarks to assess practical coverage and robustness under noisy metadata conditions. Here, ***FinalFormat*** denotes the paper's compact semantic type layer, ***SourceKeywords*** are the matched header tokens that justify each assignment, and ***HeadersIQ*** is the resulting data source-level summary metric derived from activated DQA checks.

This paper makes four main contributions:

1. Framework. We propose an explainable header-centric framework for metadata-only semantic typing that links inferred semantic types to rule-grounded data quality assessment, while a parallel KG-mapping pathway supports KG-oriented interpretation and benchmark comparison.

2. Semantic layer and traceability. We define a compact ***FinalFormat*** abstraction layer, supported by curated lexical resources and token-level traceability mechanisms to enable interpretable and auditable semantic assignments.

3. Large-scale evaluation. We provide a broad empirical study across heterogeneous benchmarks to examine the scalability, coverage, and robustness of header-centric annotation under noisy real-world metadata conditions.

4. Quality monitoring and benchmark diagnosis. We introduce ***HeadersIQ*** for data source-level quality monitoring and show how header-centric outputs can support diagnostic analysis of benchmark and ground-truth limitations in KG-oriented evaluation.

Taken together, these contributions position the framework as a scalable, explainable, metadata-only approach to semantic type annotation, provenance-aware traceability, and KG-oriented analysis.

In addition, ongoing work includes a **Statistical Audit** designed to provide correctness-oriented evidence for ***FinalFormat*** assignments beyond the coverage-focused indicators reported in this paper. A replication package, including appendices and implementation details, is publicly available at [15]. Subsequent references to supplementary materials in the paper refer to this package.

2. Related Work

2.1. Column Type Annotation (CTA) Paradigms

In this paper, the relevant setting is metadata-only CTA, where semantic inference relies on headers and related metadata rather than cell values.

Following Liu et al. [5], CTA systems can be grouped into four paradigms:

(i) heuristic dictionary-based systems that match header tokens against controlled vocabularies or KG labels, such as Wang et al. [16], C2 [17], MAGIC [18], and Alobaid et al. [19];

(ii) heuristic and iterative methods that propagate constraints or distribute semantic cues across table elements, including TableMiner+ [20], CSV2KG [21], T2K [22] and the MTab family [23, 24, 25];

(iii) feature-based machine learning approaches that use engineered features over values and/or headers, for example, Limaye et al. [26], and Mulwad et al. [27, 28]; and

(iv) deep learning and embedding-based methods, including Sherlock [4], Sato [29], TURL [30], Doduo [31], RECA [32], and DAGOBAH [33], which exploit representations learned from large table corpora. Only a small subset of systems (12 in Liu's *T0* category) deal with headers, and even fewer operate without access to cell values, leaving an underexplored region in the CTA landscape and motivating our header-centric, metadata-driven design. A broader comparative analysis, including additional recent models, is provided in **Appendix A [15]**.

2.2. Attribute-Level Data Quality Dimensions

Building on a previous framework [14], which established the link between semantic type assignments and executable validation rules, the present work extends this integration with a formally grounded DQI (Data Quality Issues) taxonomy and large-scale empirical evaluation.

Attributes (columns) constitute the primary semantic units of tabular data. Classical data quality literature identifies several core dimensions associated with attribute-level assessment [34, 35]. These include:

- **Accuracy**: the closeness between a recorded value and the corresponding real-world entity or state.
- **Completeness**: the degree to which expected values are present (absence of missing tuples or values).
- **Consistency**: the absence of violations of semantic or structural rules (e.g., type constraints, integrity constraints).
- **Uniqueness**: the absence of unintended duplicates that represent the same real-world entity.
- **Timeliness**: the validity of temporal information with respect to expected time bounds or recency.

Many data quality problems can be interpreted as violations of one or more of these dimensions. However, without semantic interpretation of attribute labels, automated profiling tools often fail to determine which constraints should apply. In this work, semantic type detection provides the missing link between metadata interpretation and dimension-aware validation.

2.3. Data Quality Issues (DQIs) and Taxonomic Grounding

To operationalise dimension-aware validation, we adopt the structured taxonomy of 22 Data Quality Issues (DQIs) compiled by Visengeriyeva and Abedjan (2020) [36]. This taxonomy maps concrete error patterns to specific Data Quality Dimensions.

Examples of DQIs particularly relevant to attribute-level validation include

- **DQI #1 – Missing Data** (Accuracy, Completeness): null values, blank entries, disguised placeholders.
- **DQI #5 – Extraneous Data** (Consistency, Uniqueness): additional unexpected content within attribute values.
- **DQI #9 – Duplicates** (Uniqueness): repeated tuples or values representing the same entity.
- **DQI #13 – Temporal Mismatch** (Accuracy, Timeliness): invalid or structurally incorrect temporal values.
- **DQI #15 – Domain Violation** (Accuracy): values violating semantic constraints of the attribute.
- **DQI #17 – Wrong Data Type** (Consistency): values violating declared or inferred data type.
- **DQI #19 – Uniqueness Violation** (Uniqueness): explicit violation of identifier-level uniqueness constraints.

Although certain DQIs (e.g., referential integrity violations across relations) require cross-table analysis and fall outside the scope of this study, a substantial subset can be deterministically activated through attribute-level semantic interpretation.

3. Methodology

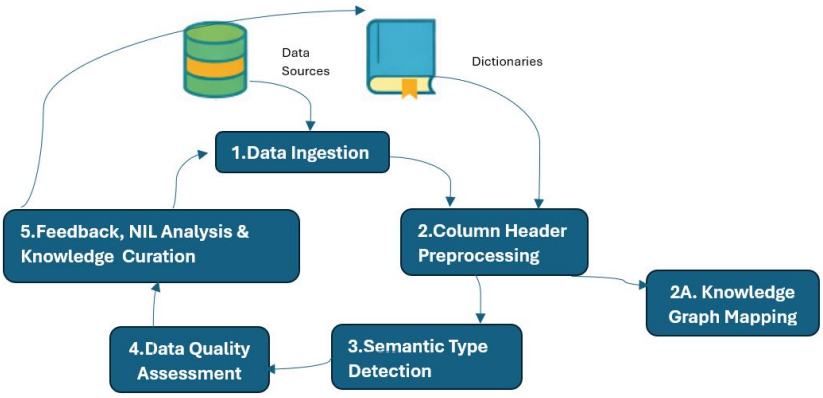

Figure 2: Pipeline for Semantic Type Detection, Knowledge Graph Mapping, and Data Quality Assessment.

Figure 2 summarises the end-to-end workflow. After preprocessing, the pipeline branches into two parallel pathways: semantic type detection, which supports downstream data quality assessment, and knowledge graph mapping, which supports KG-oriented interpretation and benchmark comparison. NIL analysis feeds back into dictionary curation.

3.1. Data Ingestion

The ingestion stage serves as the entry point for tabular data from various sources, ranging from relational databases to datasets distributed in file-based formats such as CSV files, text files, and spreadsheets, particularly those obtained from large repositories such as UCI [37] or Kaggle [38]. Because these sources vary widely in structure, ingestion handles multiple file formats and extracts initial metadata such as column headers, optional descriptions, and, when available, indicators of primary or foreign keys. Basic validation filters incomplete or malformed tables, and character encodings are normalised to ensure consistent downstream processing. The resulting metadata forms the basis for dictionary expansion and header preprocessing.

A key component of the framework is a pair of curated dictionaries:

(i) a ***formats dictionary*** (formats_dictionary.txt [15]) that maps normalised header tokens (synonyms, variants, and domain-specific terms) to 39 ***FinalFormat*** types (see Table 2, Section 3.3), and

(ii) an ***abbreviations dictionary*** (abbreviations_dictionary.txt [15]) that expands shorthand forms into terms registered in the ***formats dictionary***.

***Formats Dictionary* Creation and Growth**.

The ***formats dictionary*** [15] was initially derived from header tokens in **earlier work [14]** and later expanded with DBHeaders [39], derived from Web Data Commons [40] over more than 90 million tables in the Web Table Corpus [41]. Through iterative NIL-driven review over almost 119,000 analysed columns, it grew to roughly 2,800 entries and now supports 39 ***FinalFormat*** types. Table 1 illustrates representative mappings. These mappings ensure that semantically related or ambiguous headers are consistently normalised to interpretable ***FinalFormat*** types, such as *name, city, binary, categorical*, and *postalcode*. The coverage results reported in Sections 4.2 and 4.4 reflect the dictionary snapshot available when each benchmark was processed.

Table 1
Example mappings from terms to ***FinalFormats*** (in ***formats dictionary*** file [15])

Term	→ *FinalFormat* type	Term	→ *FinalFormat* type
id	→ *IDcolumn*	author	→ *name*
day and time	→ *datetime*	website	→ *URLformat*
boolean	→ *binary*	day of week	→ *weekday*
qualitative	→ *categorical*	address	→ *street*
height	→ *numerical>=0*	postal code	→ *postalcode*
birthplace	→ *city*	internet protocol	→ *IPformat*
territory	→ *state*	cellphone	→ *phone*

***Abbreviation Dictionary* Creation and Expansion.**
To complement the ***formats dictionary***, we maintain a dedicated ***abbreviations dictionary*** [14], [15] that contains over 1,000 mappings from shorthand or compressed forms to their expanded variants that map to terms present in the ***formats dictionary.*** This expansion is crucial for decoding terse headers (e.g., *"DOB"* → ***date*** *of birth* → *date* or *"pct"* → *percentage*). Figure 3 illustrates representative abbreviation categories, covering acronyms, misspellings (e.g., *"addre"* → *address, "phn"* → *phone,* drawn partly from NameGuess [42]), domain jargon and truncations. In this figure, the bold terms in the Expansion column correspond to the ***SourceKeywords*** that trigger the ***FinalFormat*** assignment. In every case, at least one term in the expanded form links to a ***FinalFormat*** type; for example, *CEO* → *chief executive* ***officer*** → *name.* Here the ***SourceKeyword*** is ***officer***, which is linked to ***FinalFormat*** *name* in the ***formats dictionary***.

Acronyms

Abbreviation	Expansion	FinalFormat
CEO	chief executive **officer**	name
DOB	**date** of birth	date
GPA	**grade** point average	numerical>=0

Misspellings

Abbreviation	Expansion	FinalFormat
addre	**address**	street
lettr	**letter**	string
phn	**phone**	phone

Domain jargon

Abbreviation	Expansion	FinalFormat
bp	**blood pressure**	bloodpressure
gp	**games** played	numerical>=0
vs	**versus**	binary

Truncations

Abbreviation	Expansion	FinalFormat
nbr	**number**	numerical
mo	**month**	month
wgt	**weight**	numerical>=0

Figure 3: Abbreviation categories and their expansions.

Like the main ***formats dictionary***, the ***abbreviation dictionary*** also grows iteratively through NIL analysis and expert review, supporting continual adaptation as new data sources and types of noise are encountered.

Alignment with the SemTab 2025 Challenge Priorities. This combined strategy of dictionary growth and abbreviation expansion aligns with several key priorities highlighted in SemTab 2025 [10], particularly robust alias/acronym resolution, noise resilience, NIL detection for unmappable headers, and ambiguity handling in tabular metadata.

3.2. Column Header Preprocessing

Preprocessing converts raw column headers into actionable semantic annotations using the ***FinalFormat*** type system. The main stages are the following.

1. Metadata Extraction and Normalisation. Extract header terms, normalise case and punctuation, and split compound or camelCase tokens to standardise heterogeneous naming conventions.

2. Abbreviation and Variant Handling. Expand over 1,000 abbreviations and spelling variants, enabling robust treatment of noisy, truncated, or misspelled headers commonly found in real-world data.

3. *SourceKeywords* Extraction. Extract and record the dictionary-matched trigger token(s) from the cleaned header and optional description (after abbreviation expansion) that led to the ***FinalFormat*** assignment, providing token-level traceability.

For example, in *has context, "has"* is recorded as the ***SourceKeyword*** that triggers the ***FinalFormat*** **binary**; similarly, *"bp"* expands to *"blood pressure"*, which is recorded as the ***SourceKeyword*** triggering the assignment to ***FinalFormat*** *bloodpressure.*

3.3. Semantic Type Detection

The ***FinalFormat*** system provides the semantic backbone of the framework, defining 39 interpretable types, listed in Table 2, that sit between coarse atomic data types (e.g., string, integer, date) and fine-grained ontology classes in DBpedia or Schema.org. This layer is expressive enough to support meaningful data-quality checks while remaining simple and explainable.

Definition and Scope.
FinalFormat types cover categories such as:
- **Numerical**: generic numeric, non-negative, and bounded numerical types.
- **Geographical**: *city, country, postalcode, state, street.*
- **Temporal**: *date, datetime, day, hour, month, time, week, weekday, year.*
- **Categorical**: Keywords such as "class", "status", "type", etc.
- **Name**: Keywords such as "director", "actor", "publisher", or "firstName".
- **Identifiers**: *IDcolumn* (e.g."ISBN", "SSN"), *URLformat, IPformat, phone.*
- **Binary**: Boolean fields or fields identified through cues ("has").
- **Textual**: descriptive, free text, assigned to ***FinalFormat*** string.

Table 2: The 39 ***FinalFormats***. Blue shading indicates bounded numerical ***FinalFormats***.

FinalFormat	FinalFormat	FinalFormat
acidity	heartrate	percentage
age	hour	pH
alkalinity	IDcolumn	phone
angle	IPformat	postalcode
binary	latitude	saltness
bloodpressure	longitude	state
categorical	modelname	street
city	money	string
country	month	time
date	name	URLformat
datetime	normalized	week
day	numerical	weekday
E-mailformat	numerical>=0	year

Matching and annotation. Semantic type assignment is performed through deterministic rule-based matching. Exact, substring and fuzzy matching are applied to assign ***FinalFormat*** types using the curated ***Formats*** and ***Abbreviations dictionaries. SourceKeywords*** record the supporting header or description tokens for explanation and auditing.

3.4. Knowledge Graph Mapping

Each column was linked to the most semantically aligned property in DBpedia [2] and Schema.org [3]. This mapping stage operates as a parallel pathway after preprocessing, placing header-centric metadata into a knowledge-graph vocabulary space for KG-oriented interpretation and benchmark comparison.

Figure 4 illustrates real mappings from the KG-mapping pathway, showing header normalisation, abbreviation expansion, dictionary matching, ***FinalFormat*** assignment, ***SourceKeyword*** extraction, and ontology alignment with DBpedia and Schema.org. The examples highlight how preprocessing improves semantic alignment, as abbreviated or compressed headers such as Wt, Yr, ISO, OS, NOC, and obp are expanded before mapping. Several cases produce accurate ontology matches, such as weight, year, Organization, operatingSystem, and percentage, while others reveal limitations in ontology coverage or granularity, including NIL results in Schema.org and the more specific copyrightYear returned for Yr.

These cases show that KG-based annotation can benefit from abbreviation expansion, but semantic accuracy still depends on the available vocabulary and modelling choices of the target ontology. Importantly, such KG-mapping limitations do not affect ***FinalFormat*** assignment or downstream DQA activation. After preprocessing, semantic type detection and KG mapping proceed as parallel pathways: ***FinalFormat*** assignment supports DQA, whereas KG mapping is used for ontology-oriented interpretation and benchmark comparison.

Column	Expanded Column	FinalFormat	SourceKeyword	DbpediaType	DBpedia Score	SchemaType	Schema Score
#	number	numerical	number	number	1	Number	1
ISO	international **organization** for standardization	categorical	organization	Organization	1	Organization	1
OS	**operating system**	categorical	operating system	operatingSystem	1	operatingSystem	1
Wt	weight	numerical>=0	weight	weight	1	weight	1
Yr	year	year	year	year	1	copyrightYear	0.85
Admin	administrator	name	administrator	administrator	1	NIL	0
NOC	national olympic **committee**	categorical	committee	committee	1	NIL	0
obp	on-base **percentage**	percentage	percentage	percentage	1	NIL	0

Figure 4: Examples of ***FinalFormat*** assignments and KG mappings.

Overall, ***FinalFormat*** delivers semantically richer classifications than atomic data types, while remaining more compact and operational than ontology mappings, thereby enhancing interpretability and data-quality profiling.

3.5. Data Quality Assessment

As introduced in earlier work [14], each inferred ***FinalFormat*** activates one or more predefined sets of data quality validation functions. These checks include, for example, uniqueness enforcement for identifier formats, bounded-range validation for numerical formats, and structural validation for dates and categorical fields.

In the present study, rule activation is grounded in the DQI taxonomy (Section 2.3). Each validation function yields cell-level DQI instances, which are aggregated into ***HeadersIQ*** (Subsection 3.5.1).

Representative examples include:

- `check_numerical_ge_zero`: enforces non-negative constraint (DQI #1, #15, #17).
- `check_numerical_between`: validates bounded numeric ranges (DQI #1, #15, #17).
- `check_id_attributes`: validates identifier properties (DQI #1, #9, #15, #17, #19).
- `check_date`: validates temporal structure and bounds (DQI #1, #6, #13, #15).
- `check_postal_code`: validates structural constraints (DQI #1, #15, #17).
- `check_email_format`: enforces email syntax validity (DQI #1, #15).

For bounded numerical ***FinalFormats***, `check_numerical_between` applies a predefined interval associated with the inferred semantic type. For example, a column assigned to ***FinalFormat*** *age* is validated against the adopted interval 0–130, while *pH* is validated against 0–14. The validator, therefore, does not require additional per-dataset input at runtime; it uses the interval linked to the assigned ***FinalFormat***. These bounds are maintained as configurable domain parameters linked to ***FinalFormat*** types rather than hard-coded properties of individual datasets, so they can be revised independently as domain requirements or governance policies evolve.

This mapping ensures that rule activation is not heuristic, but is grounded in a predefined taxonomy of violations aligned with classical data quality theory. Each detected violation is recorded as a cell-level DQI instance.

3.5.1. *HeadersIQ*: A Metric for Data/Information Quality Monitoring

HeadersIQ was developed as a lightweight, unweighted summary metric for monitoring data source-level quality. It reflects the proportion of evaluated cells with no detected violations under the activated data quality rules. The name ***HeadersIQ*** reflects both its foundation in column headers and its focus on Information Quality (IQ).

Formal definition. Let N_c be the number of columns, N_r the number of rows, and $\mathcal{X}$ the set of evaluated cells. Since all cells are evaluated, the total number of evaluated cells is

$$T = |\mathcal{X}| = N_c \times N_r.$$

For each $x \in \mathcal{X}$, define

$$v(x) = \begin{cases} 1, & \text{if at least one activated data quality rule detects a violation in cell } x, \\ 0, & \text{otherwise.} \end{cases}$$

Then

$$V = \sum_{x \in \mathcal{X}} v(x)$$

denotes the number of distinct cells with one or more detected violations.

The ***HeadersIQ*** score is defined as

$$\textbf{\textit{HeadersIQ}} = \left(1 - \frac{V}{T}\right) \times 100 \tag{1}$$

It produces a normalised 0-100 score representing the proportion of evaluated cells for which no violations were detected under the activated rules.

3.6. Feedback, NIL Analysis & Knowledge Curation

NIL-labelled headers are reviewed through a controlled, evidence-based human-in-the-loop feedback cycle that evolves with each benchmark iteration.

NIL Token improvement process. All columns with ***FinalFormat*** labelled as NIL follow this protocol:

- Aggregate and rank all unmatched tokens by frequency.
- Filter out meaningless or unstable entries.
- Manually review high-frequency candidates for domain relevance.
- Integrate only validated and interpretable terms into the ***formats*** and ***abbreviations dictionaries.***

4. Results and Coverage Analysis

This section presents a comprehensive evaluation of the header-centric semantic type detection framework across a diverse suite of tabular data benchmarks, including both classical STI datasets and large, real-world sources. We report coverage for ***FinalFormat*** assignments and Knowledge Graph mappings (DBpedia and Schema.org), and complement quantitative results with qualitative interpretability evidence via ***SourceKeywords***.

Sherlock [4] highlighted two main limitations of matching-based, header-dependent systems: they tend to recognise only a limited range of types, and they lack robustness when confronted with noisy or context-dependent attributes. The evaluation presented in this chapter directly addresses both points. Broad lexical coverage is demonstrated through the ***dictionaries (formats*** and ***abbreviations*)** and the 39 ***FinalFormat*** types. Robustness is demonstrated through consistently high assignment rates across heterogeneous and noisy repositories, such as Kaggle and VizNet, where normalisation, abbreviation expansion, ***SourceKeywords***, and NIL-driven feedback enhance resilience under conditions where matching-based systems have previously been shown to fail.

4.1. Data Sources and Benchmarking Scope

Scalability, robustness, and practical coverage were evaluated across seven heterogeneous benchmarks (Table 3). The evaluation spans structured repositories (**UCI [37], Prague Relational Learning Repository [43])**, large collections of web tables (**Sato** [29], **VizNet** [44]), user-generated real-world tables (**Kaggle [38])** and KG-grounded benchmarks (**SOTAB** [45], **T2Dv2** [46]).

Table 3
Structural Statistics of Benchmarks. T2Dv2 Domains (*) correspond to DBpedia classes.

Benchmark	Domains	Data Sources	Columns	Columns per Data Source
UCI	6	50	922	18.4
Prague	9	50	520	10.4
Sato	N/A	2,254	4,587	2.0
VizNet	N/A	11,199	74,915	6.7
Kaggle	17	3,364	49,727	14.8
SOTAB	N/A	N/A	737	N/A
T2Dv2	39(*)	237	1,172	4.9

Universe definitions.

Universe A - All seven benchmarks.

Universe B - Union of unique header labels extracted from Kaggle and VizNet, comprising 118,639 unique headers (after deduplicating identical header strings across both sources). Used to study distribution, long-tail behaviour and bounded-numerical coverage at scale.

4.2. Semantic Type Detection: Coverage and Diversity

We measure three aspects of semantic coverage across all benchmarks:

- ***FinalFormat*** Types: number detected in each data source group.
- ***SourceKeywords***: unique, normalized header tokens extracted across all.
- DBpedia/Schema.org Types: number of distinct ontology classes/properties successfully linked from column headers to the respective ontologies after normalisation and abbreviation expansion.

Table 4 summarises coverage. VizNet and Kaggle yield higher ***SourceKeywords*** counts due to scale; notably, Kaggle attains full ***FinalFormat*** type coverage (39/39 types).

A key result is that **only 15 core *FinalFormats*** deterministically cover the entire **76-label Sherlock/Sato space** in many-to-one mappings. Two extra labels from Sherlock do not appear in Sato (*File size and Team name*).

Table 4
Coverage of ***FinalFormat***, ***SourceKeywords***, and KG Types

Benchmark	*FinalFormat* Types	*SourceKeywords*	DBpedia Types	Schema.org Types
UCI	33	248	162	161
Prague	23	139	136	129
Sato	**15**	**76**	**76**	63
VizNet	35	**1,792**	1,045	1,010
Kaggle	**39**	**1,920**	1,046	1,138
SOTAB	23	349	246	684
T2Dv2	21	234	213	172

The consolidation preserves distinctions important for data quality (e.g., *IDcolumn, date, postalcode*), and redundant semantic variants (e.g., creator-type labels) merge naturally into compact categories, such as *name* or *categorical*.

Within **Universe B (Kaggle + VizNet)**, these 15 formats alone cover ~**80%** of the more than 118,000 headers. The distribution is heavily concentrated in a few operationally critical types (*categorical, numerical>=0, IDcolumn, string, numerical*). The complete list of these 15 types and their mappings to Sherlock/Sato is shown in **Appendix B [15]**.

Beyond the 15 core types, **Universe B** exhibits an additional 24 ***FinalFormats***, primarily bounded numerical values and domain-specific formats such as *percentage*, chemical (*ph, acidity*), health (*heartrate*), geography (*latitude/longitude*), among others. These collectively represent ~6% of headers, **totalling over 7,000 in this universe, which remain substantial in absolute terms,** and enable finer-grained Data Quality Issues (DQI) checks not supported in the Sherlock/Sato taxonomy. The complete list is also reported in **Appendix B [15]**.

Across **Universe B, NIL is approximately 13 percent (thus Valid is ~87 percent)**. Here, NIL denotes headers that did not match any ***FinalFormat*** after normalisation and abbreviation expansion.

Scalability. The modest increase of 80% from ~2,100 dictionary entries (UCI phase [14]) to ~3,800 entries supported a growth of ~103 times in practical coverage, increasing valid assignments from ~1,000 to ~103,000 headers. Details are provided in Table 5.

Table 5
Scalability evidence

Phase	Dictionary size	Headers evaluated	Valid headers assigned
UCI baseline [14]	1,800 + 300 ~ 2,100	~1,000	~1,000 (near perfect)
Kaggle+VizNet	2,800 + 1,000 ~ 3,800	118,639	~103,000 (~87%)

Bounded numerical formats. Appendix C [15] lists all adopted bounds used in this study. These intervals support finer-grained DQI checks and help disambiguate numeric families within the ***FinalFormat*** taxonomy.

4.3. NIL Token Improvement Process

A qualitative review of the top 100 NIL cases in **Universe B** shows that most unresolved headers are extremely short codes or symbols (e.g., *no, so, ga, po*) that carry little or no semantic information and therefore fall outside any meaningful ***FinalFormat*** mapping.

A smaller subset of cases (e.g., *streak, medal, designation, msrp*) exhibits domain-specific or plausible abbreviations. These are added only when their meaning is stable, and the mapping is interpretable.

These observations confirm that the remaining NIL proportion mainly reflects low-information tokens or highly specialised terminology rather than systematic gaps. The complete ranked list of 100 NIL tokens is included in **Appendix D [15]**.

4.4. Validity, NIL Rates, and Robustness

Table 6 reports NIL counts and Valid percentages for (i) ***FinalFormat*** assignments and (ii) KG mappings to DBpedia and Schema.org across **Universe A** (all seven benchmarks).

Definition of NIL and **Valid%: NIL** indicates the absence of a match after all normalisation and expansion steps. For all systems, **Valid%** is computed as ((**Total Columns – NIL**) / **Total Columns**) × 100. A **Valid%** column in ***FinalFormat*** corresponds to a successful mapping to one of the 39 defined semantic types after all dictionary expansions. For DBpedia and Schema.org, **Valid%** measures the fraction of columns whose normalised header (after the same cleaning and abbreviation expansion steps) could be aligned to at least one ontology property/class by the KG mapping component described earlier. This KG alignment is derived directly from header to ontology label similarity and is therefore independent of the ***FinalFormat*** assignment.

Table 6
NIL Counts and Valid Percentages for ***FinalFormat***, DBpedia and Schema.org

Benchmark	Total Columns	*FinalFormat*		DBpedia		Schema.org	
		NIL	Valid%	NIL	Valid%	NIL	Valid%
UCI	922	12	98.7	433	53.0	413	55.2
Prague	520	0	100.0	94	81.9	93	82.1
Sato	4,587	0	100.0	0	100.0	423	90.8
VizNet	74,915	9,435	87.4	21,244	71.6	23,795	68.2
Kaggle	49,727	7,274	85.4	16,546	66.7	17,467	64.9
SOTAB	737	5	99.3	163	77.9	7	99.1
T2Dv2	1,172	267	77.2	305	74.0	469	60.0

Key Observations.

• ***FinalFormat*** consistently achieves the highest Valid percentages across the benchmarks, reflecting the broader lexical coverage provided by the curated dictionaries and the traceability supported by ***SourceKeywords***.

• Sato obtains 100% Valid for ***FinalFormat***, indicating full coverage of this benchmark and compatibility with the Sherlock/Sato label space.

• Sato also reaches 100% Valid for DBpedia, since its annotated semantic labels correspond directly to DBpedia classes, demonstrating strong compatibility between this benchmark and the ontology vocabulary.

• By contrast, KG mapping coverage is notably lower for UCI and Kaggle, which is consistent with their more heterogeneous headers, greater use of domain-specific vocabulary, and cases in which suitable ontology properties or classes are unavailable or similarity-based alignment yields NIL.

4.5. Handling Ambiguous Headers

Beyond straightforward headers, the system also resolves complex, abbreviated, and domain-specific cases through the combined effect of normalisation, abbreviation expansion and ***SourceKeywords*** extraction. These mechanisms enable the framework to interpret signals such as *"is", "has" (binary* cues), *"range" (categorical* context), *"birth place"* (geographical semantics) or medical abbreviations such as *"trestbps" (bloodpressure).* This avoids misclassification in cases where multiple semantic cues coexist or where the lexical form is highly compressed.

These challenging cases demonstrate the system's ability to surpass surface-level matching and produce interpretable, rule-aligned ***FinalFormats***, even in noisy or unconventional metadata. A detailed description of the rule-based preprocessing, normalisation, abbreviation expansion and KG annotation steps, together with an extended set of worked examples showing how ***SourceKeywords*** and ***FinalFormats*** are assigned to complex headers, is provided in **Appendix E (Table G) [15]**.

4.6. SemTab 2024 Metadata-to-KG Evaluation

Ground-truth note. As detailed in **Appendix F [15]**, the **SemTab-2024 Metadata-to-KG ground truth (GT)** exhibits typical large-benchmark issues, including duplicates, label-granularity mismatches and occasional errors. We therefore reported results in two layers:

1. The **official GT-strict scores**, which enable head-to-head comparison with other systems, and
2. Blinded **internal diagnostics for our own outputs**, which characterise disagreement patterns under header-centric metadata-only evidence.

Official (GT-strict) results. Using the organisers' ground truth and scripts, the system correctly annotated 63/141 columns (Hit@1 = 0.45, 95% CI ≈ [0.36, 0.53]) and 67/141 (Hit@5 = 0.48, 95% CI ≈ [0.40, 0.56]). Baselines for Adwan [47] and for CVA/ MetaLinker [48] were taken from the SemTab-2024 official report [9]. These head-to-head figures appear in **Panel A of Table 7**.

Table 7
Comparative analysis on the SemTab 2024 Metadata-to-KG track

Panel A - Official (GT-strict)

System	Hit@1	Hit@5
Ours (GT-strict)	0.45 [0.36, 0.53]	0.48 [0.40, 0.56]
Adwan (Vandemoortele et al., 2024) [47]	0.75	0.92
CVA / MetaLinker (Martorana et al., 2024) [48]	0.55	0.70

Panel B - Diagnostic (ours only)

Category	Support@1	Support@5
B1: Strict + C.1 (GT-Confirmed)	0.45 [0.36, 0.53]	0.48 [0.40, 0.56]
B2: B1 + C.2 (GT-Refinement)	0.75 [0.68, 0.82]	0.78 [0.70, 0.84]
B3: B2 + C.3 (Alternative)	0.79 [0.72, 0.85]	0.83 [0.75, 0.88]

Internal diagnostic audit (ours only). A blinded post-hoc audit of our predictions was conducted using the predefined rubric described in **Appendix F [15]**:

C.1 GT-Confirmed (same canonical entity),

C.2 GT-Refinement (ontology-consistent and more appropriate for the header),

C.3 Ontology-Consistent Alternative (plausible but not in GT), and C.4 Incorrect.

Panel B of Table 7 reports the cumulative diagnostics (B1 strict + C.1, B2 + C.2, B3 + C.3) together with **95% CI**. These figures are not revised benchmark scores and are not directly comparable to the official GT-strict results in **Panel A**. Here, C.2 GT-Refinement contributed **43/141 (30.5%)** cases and C.3 Alternatives **6/141 (4%)**. Cumulative diagnostic support rises from 0.45 under strict GT to ≈ 0.75 with C.2, and to ≈ 0.79 with C.3. This suggests that a substantial subset of GT-strict mismatches is associated with label granularity, aliasing, or ontology-selection effects, rather than with wholly implausible header-centric predictions. This audit, therefore, highlights recurring **SemTab-2024 GT** limitations under metadata-only evidence and supports the framework's use for KG-oriented benchmark diagnosis and quality analysis.

4.7. Illustrative Case Studies from Real Datasets

To demonstrate the operational behaviour of the framework, we highlight representative examples from real-world datasets.

Dataset 352 – Online Retail. The `Description` attribute, classified as `String`, triggered:

- DQI #1 (Missing Data): 1,454 blank or null values.
- DQI #17 (Wrong Data Type): one numeric value incorrectly present in a textual field.

The `CustomerID` attribute, classified as `IDcolumn`, revealed layered violations:

- DQI #1 (Missing Data): 135,080 missing values.
- DQI #9 (Duplicates): 541,830 repeated identifiers.
- DQI #19 (Uniqueness Violation): 537,536 explicit uniqueness violations.

This illustrates how identifier semantics activate simultaneous completeness and uniqueness constraints.

Dataset 45 – Heart Disease. The attribute `thalach` (maximum heart rate), classified as numerically bounded (***FinalFormat*** *heartrate*), triggered a domain violation of DQI #15 due to a value exceeding the medically plausible upper bound.

In contrast, the attribute `age`, also a numerically bounded type (***FinalFormat*** *age*), passed all validation checks with observed values within the expected range (29–77), demonstrating the correct semantic alignment.

4.8. *HeadersIQ*: Data Source-Level Data Quality Measurement

Applying the metric defined in Subsection 3.5.1, we compute data source-level quality scores across the evaluated benchmarks.

For example, Dataset 45 (303 rows, 14 columns; $T = 4242$) contains $V = 7$ cells with at least one detected violation:

$$\textbf{\textit{HeadersIQ}} = \left(1 - \frac{7}{4242}\right) \times 100 \approx 99.83$$

Scores close to 100 indicate few detected violations, while lower values reflect increasing levels of detected structural, semantic, or representational issues. Because the metric is normalised by data source size, it enables direct comparison across heterogeneous tabular sources.

HeadersIQ supports quality-aware data preparation by enabling ranking and monitoring of tabular sources prior to knowledge graph integration. In this sense, it provides a lightweight governance signal relevant to data source-level reliability in settings such as **SemTab 2024 "IsGold?"** and related KG quality assessment work, including **KG2Tables**.

Representative examples and the full results table are provided in **Appendix G [15]**, together with implementation details in the *Attribute-BasedDataQualityAssessment.ipynb* notebook [15].

5. Discussion

5.1. Reassessing Header-Centric Type Detection

Header-centric annotation has often been regarded as unreliable, partly influenced by early approaches such as Sherlock [4] and Sato [29]. Our results challenge this view. With systematic preprocessing, curated and incrementally enriched dictionaries, and abbreviation expansion, headers alone contain enough semantic structure to support precise and scalable type assignments. The 39 ***FinalFormat*** types provide a compact operational layer between primitive data types and fine-grained ontology classes: richer than atomic types, more compact than the Sherlock/Sato label space, and simple enough to trigger interpretable data quality rules, while ***SourceKeywords*** preserve finer-grained lexical evidence for auditing and KG-oriented interpretation.

These comparisons should be interpreted in light of task setting. Systems such as Sherlock and Sato were designed primarily for broader semantic type detection scenarios that may exploit richer supervision or table-value context, whereas the present framework targets metadata-only, header-centric interpretation with explicit traceability and DQA activation. The goal is therefore not to outperform value-aware systems on their native setting, but to show that interpretable header-centric typing remains practically effective at scale under metadata-only constraints.

5.2. General-Purpose Core and Domain-Specific Long-Tail Modelling

A modest dictionary expansion (approximately 80% more entries) enabled more than 100-fold growth in practical coverage, increasing valid assignments from approximately 1,000 in the UCI phase to roughly 103,000 across more than 118,000 heterogeneous headers. This suggests that NIL-driven feedback can expand lexical coverage substantially without requiring equally large growth in the dictionaries. At the same time, the 76 Sherlock/Sato types map cleanly into only 15 core ***FinalFormat*** categories. An additional 24 domain-specific types extend the taxonomy to cover real-world long-tail cases without creating an overly fragmented label space. Consistent results across UCI, Prague, Sato/VizNet, Kaggle, SOTAB, and T2Dv2 indicate that the same set of 39 ***FinalFormats*** remains broadly usable across heterogeneous benchmarks, while the parallel KG-mapping pathway maintains alignment with DBpedia and Schema.org. In this sense, the 39 ***FinalFormats*** function as a compact operational vocabulary: large enough to support interpretable DQA activation and real-world long-tail cases, yet small enough to remain auditable, stable, and reusable across diverse data sources.

5.3. Data Source-Level Quality via *HeadersIQ*

HeadersIQ provided intuitive data quality scores across sources of varying sizes, offering a scalable way to monitor detected issues in large repositories. Its design is relevant to **SemTab 2024 "IsGold?"** and **KG2Tables**-style data source reliability questions and illustrates the practical value of linking semantic typing with automated data-quality assessment.

5.4. Interpreting Ground Truth in SemTab 2024

Official **SemTab 2024 Metadata-to-KG** results yielded Hit@1 = 0.45 and Hit@5 = 0.48. A blinded header-centric audit indicates that a substantial subset of GT-strict mismatches is associated with granularity, aliasing, or ontology-selection effects. This diagnostic evidence suggests that rigid GT matching can obscure semantically plausible metadata-only predictions and supports future CTA benchmarks with multi-label or graded evaluation, together with explicit human-in-the-loop validation.

5.5. Limitations and Future Work

The main limitations are (i) dependence on meaningful headers and (ii) difficulty with long-tail terms. Several results in this paper are coverage-oriented rather than direct correctness measures; ongoing work includes a **Statistical Audit** for ***FinalFormat*** correctness and a KG vocabulary-gap analysis of frequent ***SourceKeywords*** without exact DBpedia or Schema.org terms. Future work should combine header and value signals, integrate multilingual models, and explore LLM-based disambiguation for sparse, overloaded, or over-specialised metadata

6. Conclusion

This paper presents an explainable, header-centric CTA framework for metadata-only semantic typing that combines normalization, curated dictionaries, and ***SourceKeywords*** to assign 39 ***FinalFormat*** types. The same abstraction layer supports mappings to DBpedia and Schema.org, grounding cell-level data quality checks that aggregate into the data source-level score ***HeadersIQ***.

Experiments on around 120,000 header columns from seven benchmarks, together with a diagnostic audit of the **SemTab 2024 Metadata-to-KG track**, show that the framework supports large-scale metadata-only semantic table interpretation, while also providing useful benchmark-diagnostic evidence for KG-oriented evaluation. This positions the framework as a practical contribution to metadata-driven semantic annotation, quality-aware tabular preparation, and trustworthy knowledge-graph integration. More broadly, the framework also reflects robustness concerns that remain central in recent SemTab editions, particularly under noisy, ambiguous, and incomplete tabular metadata, while also opening opportunities for multilingual and LLM-assisted extensions in future work.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT/GPT5 (OpenAI) in order to: Grammar and spelling check, Paraphrase and reword, Improve writing style, and Formatting assistance. Further, the authors used ChatGPT/GPT5 (OpenAI) to assist with the development, debugging, and explanation of code used in the research workflow. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. Shigarov, Table Understanding: Problem Overview. WIREs Data Mining and Knowledge Discovery 13(1), 2023. URL: https://wires.onlinelibrary.wiley.com/doi/10.1002/widm.1482.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: A Nucleus for a Web of Open Data. In: The Semantic Web - ISWC 2007/ASWC 2007. LNCS 4825, 2007. URL: https://www.cis.upenn.edu/~zives/research/dbpedia.pdf.

[3] R. V. Guha, D. Brickley, S. Macbeth, Schema.org: Evolution of structured data on the web, Queue 13 (2015) 10 – 37. URL: https://api.semanticscholar.org/CorpusID:27038003.

[4] M. Hulsebos, K. Hu, M. Bakker, E. Zgraggen, A. Satyanarayan, T. Kraska, C. Demiralp, C. Hidalgo, Sherlock: A Deep Learning Approach to Semantic Data Type Detection. In: Proceedings of ACM SIGKDD 2019, 2019. URL: https://dl.acm.org/doi/10.1145/3292500.3330993.

[5] J. Liu, Y. Chabot, R. Troncy, V.-P. Huynh, T. Labbe, P. Monnin, From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. Journal of Web Semantics 76, 2023. URL: https://doi.org/10.1016/j.websem.2022.100761.

[6] E. Jiménez-Ruiz, O. Hassanzadeh, V. Efthymiou, J. Chen, K. Srinivas, SemTab 2019: Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems. In: The Semantic Web – ISWC 2020. LNCS 12123, 2020. URL: https://link.springer.com/chapter/10.1007/978-3-030-49461-2_30.

[7] E. Jiménez-Ruiz, O. Hassanzadeh, V. Efthymiou, J. Chen, K. Srinivas, V. Cutrona, Results of SemTab 2020. CEUR Workshop Proc. 2775, 2020. URL: https://ceur-ws.org/Vol-2775/paper0.pdf.

[8] V. Cutrona, J. Chen, V. Efthymiou, O. Hassanzadeh, E. Jiménez-Ruiz, J. Sequeda, K. Srinivas, N. Abdelmageed, M. Hulsebos, D. Oliveira, C. Pesquita, Results of SemTab 2021. CEUR Workshop Proc. 3103, 2021. URL: https://ceur-ws.org/Vol-3103/paper0.pdf.

[9] O. Hassanzadeh, N. Abdelmageed, M. Cremaschi, V. Cutrona, F. D'Adda, V. Efthymiou, B. Kruit, E. Lobo, N. Mihindukulasooriya, N. Pham, Results of SemTab 2024. CEUR Workshop Proc. 3889, 2024. URL: https://ceur-ws.org/Vol-3889/paper0.pdf.

[10] M. Cremaschi, F. D'Adda, F. Jiomekong Azanzi, J. Petit Yvelos, E. Jiménez-Ruiz, O. Hassanzadeh, SemTab 2025: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching., 2025. URL: https://sem-tab-challenge.github.io/2025/.

[11] N. Abdelmageed, SemTab 2024 - IsGold? Track, 2024. URL: https://sem-tab-challenge.github.io/2024/tracks/is-gold-track.html.

[12] N. Abdelmageed, E. Jiménez-Ruiz, O. Hassanzadeh, B. König-Ries, KG2Tables: A Domain-Specific Tabular Data Generator to Evaluate Semantic Table Interpretation Systems. Transactions on Graph Data and Knowledge 3(1), 2025. URL: https://drops.dagstuhl.de/storage/08tgdk/tgdk-vol003/tgdk-vol003-issue001/TGDK.3.1.1/TGDK.3.1.1.pdf.

[13] M. Hulsebos, P. Groth, C. Demiralp, AdaTyper: Adaptive Semantic Column Type Detection, 2023. URL: https://arxiv.org/abs/2311.13806.

[14] M. V. Silva, H. Herrmann, V. Maxville, Attribute-based semantic type detection and data quality assessment, in: Proceedings of the 11th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT 2024), Sharjah, United Arab Emirates, 2024, pp. 119–124. URL: https://doi.org/10.1109/BDCAT63179.2024.00030.

[15] M. V. Silva, All codes and files developed in this paper, 2026. URL: https://github.com/HeadersIQ/HeadersIQ-main/tree/main, GitHub repository.

[16] J. Wang, H. Wang, Z. Wang, K. Zhu, Understanding Tables on the Web. In: Conceptual Modeling (ER 2012). LNCS 7532, 2012. URL: https://link.springer.com/chapter/10.1007/978-3-642-34002-4_11.
[17] U. Khurana, S. Galhotra, Semantic Annotation for Tabular Data, 2020. URL: https://arxiv.org/abs/2012.08594.
[18] B. Steenwinckel, F. De Turck, F. Ongenae, MAGIC: Mining an Augmented Graph Using INK, 2021. URL: https://ceur-ws.org/Vol-3103/paper6.pdf.
[19] A. Alobaid, O. Corcho, Balancing Coverage and Specificity for Semantic Labelling of Subject Columns. Knowledge-Based Systems 240, 2022. URL: https://doi.org/10.1016/j.knosys.2021.108092.
[20] Z. Zhang, Effective and Efficient Semantic Table Interpretation Using TableMiner+. Semantic Web 8(6), 2017. URL: https://journals.sagepub.com/doi/full/10.3233/SW-160242.
[21] B. Steenwinckel, G. Vandewiele, F. De Turck, F. Ongenae, CSV2KG: Transforming Tabular Data into Semantic Knowledge. In: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2019). CEUR Workshop Proc. 2553, 2019. URL: https://ceur-ws.org/Vol-2553/paper5.pdf.
[22] D. Ritze, O. Lehmberg, R. Meusel, C. Bizer, Matching HTML Tables to DBpedia. In: Proceedings of WIMS 2015, 2015. URL: https://dl.acm.org/doi/10.1145/2797115.2797118.
[23] P. Nguyen, N. Kertkeidkachorn, R. Ichise, H. Takeda, MTab: Matching Tabular Data to Knowledge Graph Using Probability Models. In: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2019). CEUR Workshop Proc. 2553, 2019. URL: https://arxiv.org/abs/1910.00246.
[24] P. Nguyen, I. Yamada, N. Kertkeidkachorn, R. Ichise, H. Takeda, MTab4Wikidata at SemTab 2020: Tabular Data Annotation with Wikidata. In: Proceedings of SemTab 2020. CEUR Workshop Proc. 2775, 2020. URL: https://ceur-ws.org/Vol-2775/paper9.pdf.
[25] P. Nguyen, I. Yamada, N. Kertkeidkachorn, R. Ichise, H. Takeda, SemTab 2021: Tabular Data Annotation with MTab Tool. In: Proceedings of SemTab 2021. CEUR Workshop Proc. 3103, 2021. URL: https://ceur-ws.org/Vol-3103/paper8.pdf.
[26] G. Limaye, S. Sarawagi, S. Chakrabarti, Annotating and Searching Web Tables Using Entities, 2010. URL: https://dl.acm.org/doi/10.14778/1920841.1921005.
[27] V. Mulwad, T. Finin, A. Joshi, Semantic Message Passing for Generating Linked Data from Tables. In: Proceedings of ISWC 2013. LNCS 8218, 2013. URL: https://dl.acm.org/doi/10.1007/978-3-642-41335-3_23.
[28] V. Mulwad, T. Finin, Z. Syed, A. Joshi, Using Linked Data to Interpret Tables. In: Proceedings of the 1st International Workshop on Consuming Linked Data (COLD 2010). CEUR Workshop Proc. 665, 2010. URL: https://ceur-ws.org/Vol-665/MulwadEtAl_COLD2010.pdf.
[29] D. Zhang, Y. Suhara, J. Li, M. Hulsebos, Ç. Demiralp, W.-C. Tan, Sato: Contextual Semantic Type Detection in Tables. Proc. VLDB Endow. 13(11), 2020. URL: https://www.vldb.org/pvldb/vol13/p1835-zhang.pdf.
[30] X. Deng, H. Li, H. Shi, H. Zhang, J. Zhou, Y. Zhao, H. Chen, H. Peng, K. Chen, X. Chen, J. Tang, TURL: Table Understanding through Representation Learning. Proc. VLDB Endow. 14(3), 2020. URL: https://www.vldb.org/pvldb/vol14/p307-deng.pdf.
[31] Y. Suhara, J. Li, Y. Li, D. Zhang, C. Demiralp, C. Chen, W.-C. Tan, Annotating Columns with Pre-trained Language Models. In: Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), 2022. URL: https://doi.org/10.1145/3514221.3517906.
[32] Y. Sun, H. Xin, L. Chen, RECA: Related Tables Enhanced Column Semantic Type Annotation Framework. Proc. VLDB Endow. 16(6), 2023. URL: https://doi.org/10.14778/3583140.3583149.
[33] Y. Chabot, T. Labbé, J. Liu, R. Troncy, DAGOBAH: An End-to-End Context-Free Tabular Data Semantic Annotation System. In: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2019). CEUR Workshop Proc. 2553, 2019. URL: https://www.cs.ox.ac.uk/isg/challenges/sem-tab/2019/papers/DAGOBAH.pdf.
[34] C. Batini, M. Scannapieco, Data and Information Quality: Dimensions, Principles and Techniques, Springer, 2016. URL: https://link.springer.com/book/10.1007/978-3-319-24106-7.
[35] T. Dasu, T. Johnson, Exploratory Data Mining and Data Cleaning, Wiley, 2003. URL: https://onlinelibrary.wiley.com/doi/epdf/10.1002/0471448354.
[36] L. Visengeriyeva, Z. Abedjan, A Taxonomy of Dirty Data, 2020. URL: https://dl.acm.org/doi/pdf/10.

1145/3371925.
[37] M. Kelly, R. Longjohn, K. Nottingham, The UCI Machine Learning Repository. University of California, 2025. URL: https://archive.ics.uci.edu.
[38] Kaggle, Kaggle Datasets, 2025. URL: https://www.kaggle.com/datasets.
[39] W. D. Commons, DBheaders.txt file, 2014. URL: https://data.dws.informatik.uni-mannheim.de/webtables/2014-02/statistics/DBheaders.txt.
[40] O. Lehmberg, D. Ritze, R. Meusel, C. Bizer, A Large Public Corpus of Web Tables Containing Time and Context Metadata. In: Proceedings of the 25th International Conference Companion on World Wide Web (WWW '16 Companion), 2016. URL: https://doi.org/10.1145/2872518.2889386.
[41] M. Cafarella, A. Halevy, D. Wang, E. Wu, Y. Zhang, WebTables: Exploring the Power of Tables on the Web. Proc. VLDB Endow. 1(1), 2008. URL: https://yz.mit.edu/old-site/papers/webtables-vldb08.pdf.
[42] J. Zhang, Z. Shen, B. Srinivasan, S. Wang, H. Rangwala, G. Karypis, NameGuess: Column Name Expansion for Tabular Data. In: Proceedings of EMNLP 2023, 2023. URL: https://aclanthology.org/2023.emnlp-main.820.pdf.
[43] J. Motl, O. Schulte, The Prague Relational Learning Repository. In: Proceedings of the 25th International Conference on Inductive Logic Programming (ILP 2015). LNCS 9616, 2015. URL: https://arxiv.org/html/1511.03086v2.
[44] K. Hu, N. Gaikwad, M. Bakker, M. Hulsebos, E. Zgraggen, C. Hidalgo, T. Kraska, G. Li, A. Satyanarayan, C. Demiralp, VizNet: Towards a Large-Scale Visualization Learning and Benchmarking Repository, 2019. URL: https://dl.acm.org/doi/10.1145/3290605.3300892.
[45] K. Korini, R. Peeters, C. Bizer, SOTAB: The WDC Schema, 2022. URL: https://ceur-ws.org/Vol-3320/paper1.pdf.
[46] D. Ritze, C. Bizer, Matching Web Tables to DBpedia - A Feature Utility Study. In: Proceedings of the 20th International Conference on Extending Database Technology (EDBT 2017), 2017. URL: https://openproceedings.org/2017/conf/edbt/paper-148.pdf.
[47] N. Vandemoortele, B. Steenwinckel, S. Van Hoecke, F. Ongenae, Scalable Table-to-Knowledge Graph Matching from Metadata Using LLMs, 2024. URL: https://ceur-ws.org/Vol-3889/paper4.pdf.
[48] M. Martorana, X. Pan, B. Kruit, T. Kuhn, J. van Ossenbruggen, Column Vocabulary Association (CVA): Semantic Interpretation of Dataless Tables, 2024. URL: https://sem-tab-challenge.github.io/2024/semtab2024-proceedings/paper2.pdf.