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

ABSTRACT

Semantic Table Interpretation (STI) links tabular data to knowledge graphs; within it, **Column Type Annotation (CTA)** assigns meaningful semantic types to columns to enable interoperability and consistent mapping to ontology classes and properties. However, most existing approaches still treat column headers as secondary signals, limiting explainability and large-scale generalisation.

We present an **explainable, header-centric CTA framework** built on ***FinalFormat***, a curated set of 39 actionable types. The method combines evolving dictionaries, normalization, and semantic similarity with a complementary ML stage for ambiguous or unseen headers; ***SourceKeywords*** provide token-level **explainability** for every type assignment.

Building on an initial **UCI phase (~1k columns)** with dictionaries of 1,8k format terms and 300 abbreviations, we expanded them by **~80%** (to 2,8k and 1,000, respectively) via NIL-driven manual feedback and external corpora such as NameGuess. With these modest expansions, the framework generalised to the **Kaggle + VizNet** superset (**≈118k headers**), achieving **≈87%** valid **header-only** assignments (**≈103 k/118 k**), a **≈103× coverage efficiency** relative to the UCI baseline. Roughly half of these assignments are unambiguous dictionary matches; the remainder involve multi-word headers currently under validation. Evaluation across **Prague**, **T2Dv2**, **SOTAB**, and **SemTab-2024 Metadata-to-KG confirms robustness**, with mappings to **DBpedia** and **Schema.org**.

The 76 Sherlock/SATO types compress into only 15 core ***FinalFormats***, extended by 23 other domain-specific types. On SemTab-2024, GT-strict results (**Hit@1/5 = 0.45/0.48**) rise to **0.80/0.83** under human reconciliation.

We also introduce ***HeadersIQ***, a dataset-level data quality metric (0–100) aligned with SemTab 2024's IsGold? track and KG2Tables. Altogether, the framework satisfies emerging SemTab 2025 requirements (noise robustness, alias/acronym resolution, NIL handling) and enables trustworthy, large-scale, header-centric CTA. All code, dictionaries, models, and benchmark subsets are released for full reproducibility.

CCS CONCEPTS

• Information systems → Data management systems → Information integration → Data cleaning

• Computing methodologies → Knowledge representation and reasoning; Machine learning

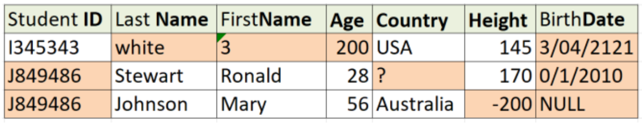
KEYWORDS

Semantic table interpretation, column type annotation, data quality assessment, table understanding.

1 INTRODUCTION

In recent years, the rapid expansion of open data repositories and domain-specific tabular datasets has made semantic table understanding a central challenge for data integration and analytical interoperability. Yet most real-world tables remain poorly annotated or lack explicit semantic types, making semantic integration both difficult and error-prone.

Semantic Table Interpretation (STI), mapping tables to knowledge graphs or ontologies, enables these tasks. The **Column Type Annotation (CTA)** subtask, which assigns meaningful types to table columns, is pivotal for interoperability, data cleaning, and analytics. (Here, "table" denotes any rectangular column–row structure, e.g., databases, spreadsheets, UCI, or Kaggle datasets.)



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 a dirty dataset

Figure 1 illustrates typical data quality problems - duplicates, missing values, numeric outliers, formatting inconsistencies, and type violations - which can be detected by simply analyzing the words and terms in column headers. Header terms (**in bold**) such as ***ID, Name, Age, Humidity,*** and ***Date*** encode strong semantic expectations. This system extracts and normalizes these signals using continually evolving **dictionaries** (**2,800 terms to formats, 1,000 abbreviations to terms**), mapping them to 39 curated ***Final Format*** types, each linked to corresponding data-quality rules. This enables automated, explainable, and scalable data profiling, **even before value-based analysis**. At scale, across 118,000 columns, header-only analysis correctly identified the semantic type in over 86% of cases without checking sensitive cell values, indicating that the phenomenon exemplified by ***Figure 1*** is widespread rather than anecdotal.

Historical context and skepticism.

Influential works such as ***Sherlock*** (2019) [14] and ***Sato*** (2020) [45], which also introduced widely used benchmarks, discouraged header-centric features, arguing they (i) handle only simple types, (ii) fail to generalize to complex/context-dependent semantics, and (iii) depend on static, incomplete dictionaries. This shaped community norms and commercial practice (e.g., Tableau), where unrecognized headers fall back to primitive types, sacrificing semantic richness and explainability.

Emerging evidence for headers.

A recent survey by Liu et al. (2023) [23] shows many competitive CTA systems now leverage headers via lookups, features, iterative matching, or deep learning: *"**The table***

header often directly explains the contents of the column*. Making full use of the information from the table header can help to find the column type or properties more efficiently".* Since 2019, **The Semantic Web Challenges on Tabular Data to Knowledge Graph Matching (SemTab)** [16, 17, 8, 12, 7] reinforced this shift: the 2024 edition [12] introduced a **Metadata-only** track which promoted header-centric annotation for privacy and scalability, while the **IsGold?** Track [1], emphasized automated **data quality assessment** (structural noise, annotation inconsistencies, NIL misuse). Abdelmageed et al. (2025) [2] further highlight data quality as an open gap (diversity, annotation accuracy/clarity, structural coherence). Notably, ***AdaTyper* (2023) [15]**, from the Sherlock/Sato group, prioritizes header analysis before cell-based fallback, validating robust header-driven pipelines.

Motivations for a header-driven pipeline.

•**Explainability:** headers are compact, human-readable signals; assignments are traceable via SourceKeywords and dictionary mappings, supporting human-in-the-loop review.

• **Scalability, efficiency, and privacy:** analysing headers alone enables large-scale, metadata-only semantic type detection without accessing sensitive cell values; this facilitates efficient, subsequent privacy-aware data quality assessment by focusing checks on the most relevant issues for each inferred type.

• **Early detection and quality control:** many potential data quality issues, such as duplicate IDs, missing values, out-of-range dates, or categorical inconsistencies, can be anticipated even before cell-value analysis, simply by interpreting headers.

• **Empirical success:** Chapter 3 reports high coverage and accuracy across diverse datasets, often surpassing header-agnostic or value-only approaches.

Addressing classical critiques.

This approach directly overcomes the classic limitations raised by *Sherlock* [14] and its followers:

• **Static vs. Dynamic dictionaries:** An evolving 2,800-term ***formats dictionary*** is continuously expanded through feedback and error analysis to handle novel headers.

• **Ambiguity & variants:** Normalization, tokenization, and 1,000 abbreviation mappings handle misspellings, acronyms, domain jargon and truncations.

• **Beyond simple types:** The ***FinalFormat*** system (Section 2.4) bridges atomic types and ontological classes, linking each to relevant data-quality rules.

• **Optional ML classifier and NIL Assignment:** Ambiguous or unmappable headers trigger an optional ML classifier; columns still unresolved are explicitly labelled NIL, avoiding silent misclassification.

Approach and Contributions.

This work directly addresses the limitations of traditional skepticism and static dictionaries. The dynamic, feedback-driven pipeline demonstrates that robust header-centric methods are not only competitive with, and often superior to, cell-content-based or header-agnostic alternatives, establishing an explainable paradigm for STI and automated data quality.

This paper substantially extends a prior attribute/column-based semantic type detection and data quality assessment framework [37], which was originally validated on 50 datasets from the UCI Machine Learning Repository [24] with smaller dictionaries. It introduces major advances in type system coverage, dataset diversity, feedback-driven expansion, benchmarking, explainability, and automated quality metrics.

Key Contributions.

Compared to [37], this paper makes the following substantially extended contributions:

1. **Broader Benchmarking (*Table 2*)**: The framework scales from the 50 UCI datasets used previously (~1,000 columns) to more than 120,000 columns across multiple new repositories, a greater than 100 times expansion in benchmarking scope. Evaluation now included Prague, T2Dv2, VizNet/Sato, Kaggle, SOTAB, and the SemTab 2024 Metadata-to-KG track, with all sources annotated using both ***FinalFormat*** types and Knowledge Graph properties (DBpedia, Schema.org).
2. **Dynamic Dictionary Expansion and Scaling Efficiency (*Table1, Figure 3, Table 6 and Section 4.2*):** Despite only ~80% growth in the dictionaries (format terms **1,800 → 2,800**; abbreviations **300 → 1,000,** the system generalised effectively to the Kaggle+VizNet superset (**118,639 headers**). With this modest expansion, **≈87% of headers received an assigned FinalFormat label** (≈103,000/118,639), corresponding to **≈103×** coverage capacity relative to the initial UCI phase (~1,000 columns). Ongoing analysis indicates **roughly half** of these assigned labels are **direct, unambiguous dictionary matches**, while the **remaining ≈51,500 multi-word headers** are **under manual/ statistical validation**, defining the next research step.
3. **Explainability through *SourceKeywords (Figure 5 and Table 8)*:** Each semantic type assignment is linked to the specific header tokens that triggered it, enabling human-in-the-loop review, machine-learning consistency checks, and feedback-driven dictionary enrichment.
4. **Expanded Semantic Type System (*Tables 4 and 5*):** The ***FinalFormat*** taxonomy grew from 23 to 39 interpretable types, including new bounded numerical formats with explicit ranges, such as *acidity, angle, alkalinity, blood pressure*, *heartrate*, and *saltness*.
5. **Integrated Machine Learning Pipeline (*Figures 6-9*):** Five classifiers (Random Forest, Logistic Regression, Gradient Boosting, KNN, Linear SVC) were implemented on 100 data sources (50 UCI + 50 Prague) to complement rule-based detection on ambiguous/unseen cases.
6. **Benchmark-Driven Error Analysis and Ground-Truth Re-evaluation (*Table 9* and *Section 4.7*):** A detailed comparison on the **SemTab 2024 Metadata-to-KG Track**, conducted alongside CTA systems such as

Adwan and CVA/MetaLinker, revealed annotation ambiguities and ground truth inconsistencies. Through human-in-the-loop reconciliation, our system's apparent accuracy improved from 0.45/0.48 to 0.80/0.83 (Hit@1/5), highlighting that benchmark evaluation must consider semantic appropriate-ness rather than rigid label matching.

This contributes to more transparent and realistic benchmarking standards for STI.

7. ***HeadersIQ* (*Table 10*):** A lightweight dataset-level metric summarizing the proportion of clean data items (0–100), enabling scalable, comparable data quality monitoring and aligning with SemTab 2024 "IsGold?".
8. **Comprehensive Benchmark Alignment (Section 4.3):** The framework fully generalises the *Sherlock*/*Sato* taxonomy (76 labels) into only 15 core ***FinalFormat*** types while extending coverage with 23 additional domain-specific types (e.g., chemistry, geolocation, health). This confirms complete containment of existing benchmarks within our ontology and demonstrates robust generalisation to noisier, real-world repositories.

In summary, this framework directly addresses the key challenges for robust column header interpretation in semantic table annotation, as defined by the latest community benchmarks. By leveraging dictionary expansion, explainable SourceKeywords, and an optional machine learning classifier, the approach excels in both clean and noisy header scenarios, setting a new standard for scalable, resilient, and explainable column type annotation and efficient data quality assessment.

All supporting artefacts, including code, dictionaries, preprocessing scripts, rule-based functions, trained model artefacts, and benchmark subsets used in this study, are publicly available in [38]. These materials correspond exactly to the configurations used for the reported results, ensuring full reproducibility of all analyses presented in this paper.

2 METHODOLOGY

Figure 2 summarizes our end-to-end pipeline for semantic type detection and annotation.

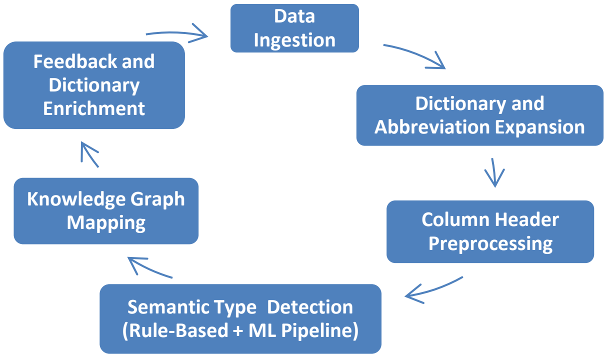

Data Ingestion
Dictionary and Abbreviation Expansion
Column Header Preprocessing
Semantic Type Detection (Rule-Based + ML Pipeline)
Knowledge Graph Mapping
Feedback and Dictionary Enrichment

Figure 2 - Pipeline for Semantic Type Detection. The process operates mainly in a header-only mode; feedback from NIL cases supports iterative dictionary refinement.

2.1 Data Ingestion

The data ingestion stage serves as the entry point for tabular data from diverse sources, including relational databases, spreadsheets, CSV files, and main tables within larger datasets such as UCI [24] or Kaggle [18] repositories. Due to the heterogeneous nature of these sources, ingestion handles multiple file formats and data structures, extracting initial metadata such as column headers, possible descriptions or information regarding Primary Keys or Foreign keys. This stage performs preliminary validation, filtering out incomplete or malformed tables, and standardizes encoding and file formats to ensure consistent downstream processing. The ingested data, together with its extracted metadata, forms the basis for dictionary expansion and header preprocessing.

2.2 Dictionary and Abbreviation Expansion

A key component of the framework is a continuously evolving ***formats dictionary*** **[38]** that maps header tokens (including synonyms, variants, and domain-specific terms) to a curated set of **39 FinalFormat** types (see *Section 2.4*). This resource underpins robust, explainable semantic type detection by normalising diverse column headers and ensuring consistent mapping across domains.

***Format Dictionary* Creation and Expansion**

The ***formats dictionary*** was first developed in previous work [37] using header tokens that could be readily associated with semantic formats observed in the initial datasets analysed. To strengthen coverage and consistency, terms from the **Web Data Commons project [21]** were later incorporated; this corpus analysed **more than 90 million tables in the Web Table Corpus [5]** and linked column headers to DBpedia entities via the **DBHeaders file** [9]. From this process an initial dictionary of roughly **1,100** unique header terms was obtained. The resource then expanded to **1,800** entries (through **ChatGPT-based term suggestions** and **manual curation**) in the version reported in **[37]**, and subsequently to **2,800 entries** through manual curation and systematic **feedback from ≈ 120,000 analysed columns** drawn from thousands of data sources. During annotation, **NIL-labelled headers** (those lacking an existing mapping) were reviewed and their new or variant tokens were incorporated into the dictionary, ensuring continual enrichment and broader lexical coverage. The resulting dictionary supports all **39 FinalFormat** semantic types, examples of which are shown in **Table 1**.

Table 1 - Example mappings from Header terms to FinalFormats (in *formats dictionary file* [38])

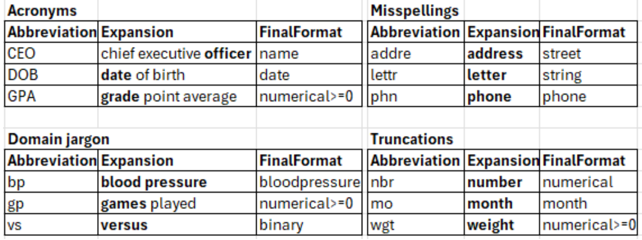
Term->FinalFormat type	Term->FinalFormat type
id->IDcolumn	author->name
day and time->datetime	website->URLformat
numeric->numerical	anniversary->date
boolean->binary	day of week->weekday
qualitative->categorical	address->street
height->numerical>= 0	postal code->postalcode
letter->string	e mail->E-mailformat
birthplace->city	internet protocol->IPformat
territory->state	cellphone->phone

Table 1 shows how semantically related or ambiguous headers are mapped to interpretable FinalFormat types like 'name', '*city*', '*IPformat*', '*categorical*', '*binary*', and '*postalcode*'. This mapping is vital for robust interpretation of noisy, abbreviated, or variant headers, as increasingly required by recent STI benchmarks.

Abbreviation Dictionary and Variant Handling

To complement the ***formats dictionary***, we maintain a dedicated ***abbreviations dictionary*** containing over 1,000 abbreviations mapping shorthand terms to their expanded forms within the ***formats dictionary***. This expansion is crucial for decoding common but terse column headers (e.g., "*DOB*" → "***Date*** *of Birth*" →"***date***" or "*pct*" straight to "***percentage***").

Figure 3 shows examples from the ***abbreviations dictionary***, illustrating the breadth and diversity of normalized forms and their mappings, including coverage of **acronyms, misspellings** (such as "addre" → "address", "phn" → "phone" derived from ***NameGuess*****[46]**), **domain jargon and truncations.** In all cases, a word in the expansion will be found in the formats dictionary, e.g., ceo -> chief executive **officer** -> ***name***.



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

The ***abbreviation dictionary***, like the main ***formats dictionary***, is expanded iteratively through feedback loops and expert review, supporting continuous adaptation as new datasets and errors are encountered.

Alignment with the SemTab 2025 Challenge Priorities.

Our approach to dictionary and abbreviation expansion directly targets the core objectives of the SemTab 2025 challenge [7], specifically, **robust alias/acronym resolution, noise resilience**, NIL detection for unmappable headers, and ambiguous metadata interpretation.

2.3 Column Header Preprocessing

Preprocessing and Semantic Type Assignment: Summary Workflow.

Our semantic type detection pipeline employs a multi-stage preprocessing and mapping process that converts raw column headers into actionable semantic annotations through the FinalFormat type system. The key stages are:

1. Metadata Extraction & Normalization

Extract core header terms and descriptions, normalize case and punctuation, and split compound or camelCase tokens to standardize diverse naming conventions.

2. Abbreviation & Variant Handling

Expand over 1,000 abbreviations and spelling variants, robustly handling noisy, truncated, or misspelled headers encountered in real-world data.

3. SourceKeywords Extraction

Identify core semantic tokens (e.g., "*age*", "*id*", "*amount*") and contextual cues (e.g., "*has*", "*is*", "*bp*") from both header names and descriptions. Unlike prior approaches that treat column typing as a black box, this framework explicitly records SourceKeywords, providing transparent evidence for each assignment. This mechanism is a key contribution of our work, enabling explainability, consistency checks, and dictionary enrichment when NIL cases occur.

2.4 Semantic Type Detection System

The ***FinalFormat*** system forms the semantic backbone of this framework, defining a compact yet expressive taxonomy of **39 interpretable types** that bridge the gap between low-level atomic datatypes (e.g., *string, integer, date*) and highly granular ontologies such as **DBpedia** and **Schema.org**. It enables **scalable, explainable, and actionable** semantic classification of tabular attributes, directly supporting automated data-quality assessment.

Rationale.

Atomic types are too coarse for detailed validation, while ontology classes are often inconsistent and difficult to operationalise. FinalFormat provides a **middle layer**, rich enough for meaningful data quality profiling but simple enough to maintain interpretability and uniform rule enforcement.

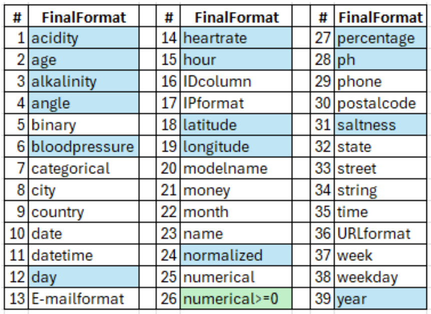
Core Principles.

- **Coverage**: captures the most frequent and operationally relevant semantic patterns across heterogeneous data domains and sources.
- **Manageability**: maintains a concise, non-fragmented set of types, reducing ambiguity and annotation overhead.
- **Actionability**: links each type to domain-specific quality rules (e.g., range, uniqueness, format checks).
- **Extensibility**: evolves through feedback-driven updates based on NIL analysis and domain expansion.

Definition and Scope.

FinalFormat types cover categories such as:

- **Numerical**: generic numeric, non-negative, and bounded numericals (in blue in ***Figure 4***, e.g., percentage, age, bloodpressure, pH, latitude, etc.).
- **Geographical**: city, country, postalcode, state, street.
- **Temporal**: date, datetime, day, hour, month, time, week, weekday, year.
- **Categorical/Qualitative**: Keywords such as "class", "label", "status", "type", etc.
- **Name**: Keywords such as "director", "actor", "publisher", or "firstName", "Full_name".
- **Identifiers**: IDColumn (e.g. "ISBN", "SSN"), URLformat, IPformat, E-mailformat, phone.
- **Binary**: Boolean fields identified through cues ("has").
- **Textual**: descriptive, free text, assigned to string.



#	FinalFormat	#	FinalFormat	#	FinalFormat
1	acidity	14	heartrate	27	percentage
2	age	15	hour	28	ph
3	alkalinity	16	IDcolumn	29	phone
4	angle	17	IPformat	30	postalcode
5	binary	18	latitude	31	saltness
6	bloodpressure	19	longitude	32	state
7	categorical	20	modelname	33	street
8	city	21	money	34	string
9	country	22	month	35	time
10	date	23	name	36	URLformat
11	datetime	24	normalized	37	week
12	day	25	numerical	38	weekday
13	E-mailformat	26	numerical>=0	39	year

Figure 4 – The 39 FinalFormats

Matching and Annotation Algorithms.

The semantic type assignment process combines deterministic rule-based matching with a robust machine learning (ML) pipeline to maximize accuracy, coverage, and explainability.

1. Rule-Based Matching.

We first apply exact, substring, and fuzzy matching techniques to assign ***FinalFormat*** semantic types and link to knowledge graph properties. This stage leverages large, curated *dictionaries* of formats and *abbreviations*.

2. Machine Learning Pipeline.

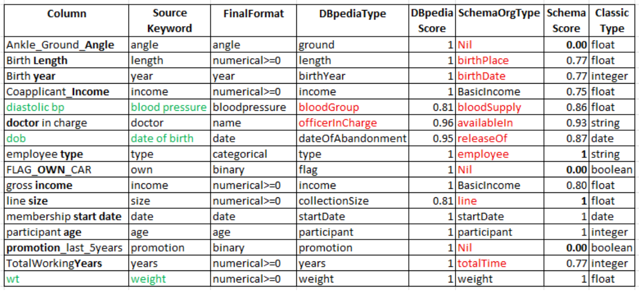
For all non-ID columns, we apply a machine-learning pipeline to cleaned and abbreviation-expanded headers (and optional descriptions). The rule-based stage contributes explicit features (e.g., *SourceKeywords*) and supports reconciliation, while classifiers systematically assign the semantic type across all columns. Headers that cannot be confidently mapped are explicitly assigned a NIL type and logged for manual review and dictionary enrichment, reducing unknown rates and increasing resilience to ambiguous or missing metadata.

Full implementation details, including preprocessing, feature extraction, model selection, and evaluation, are provided in ***Appendix A*** [38] (*notebook Attribute-BasedMLSemanticType Detection.ipynb*) [38], which also contains ***Algorithm 1*** and the complete reproducibility configuration (fixed seed 42, 80/20 split, SMOTE, GridSearchCV 3-fold).

2.5 Knowledge Graph Mapping

Each FinalFormat is linked to the most semantically aligned property in **DBpedia [4] and Schema.org [20],** using **FastText embeddings** and **cosine similarity** to compare column headers with ontology labels, capturing subtle lexical nuances and improving alignment beyond exact string matches.

Figure 5 illustrates real mappings, showing the full pipeline of header normalisation, abbreviation expansion, dictionary matching, *FinalFormat* assignment, and *SourceKeyword* extraction, alongside ontology alignment results. The examples highlight both accurate matches and areas of limited ontology coverage, for instance, *bloodGroup* and *officerInCharge* in **DBpedia**, or several **Schema.org** terms that were incorrectly aligned despite high similarity scores (shown in red). These issues reveal that although KG-based annotation benefits from abbreviation expansion (e.g., *bp*, *dob* and *wt*), semantic accuracy is not always guaranteed.



Column	Source Keyword	FinalFormat	DBpediaType	DBpedia Score	SchemaOrgType	Schema Score	Classic Type
Ankle_Ground_**Angle**	angle	angle	ground	1	Nil	**0.00**	float
Birth **Length**	length	numerical>=0	length	1	birthPlace	0.77	float
Birth **year**	year	year	birthYear	1	birthDate	0.77	integer
Coapplicant_**Income**	income	numerical>=0	income	1	BasicIncome	0.75	float
diastolic bp	blood pressure	bloodpressure	bloodGroup	0.81	bloodSupply	0.86	float
doctor in charge	doctor	name	officerInCharge	0.96	availableIn	0.93	string
dob	date of birth	date	dateOfAbandonment	0.95	releaseOf	0.87	date
employee **type**	type	categorical	type	1	employee	**1**	string
FLAG_**OWN**_CAR	own	binary	flag	1	Nil	**0.00**	boolean
gross **income**	income	numerical>=0	income	1	BasicIncome	0.80	float
line **size**	size	numerical>=0	collectionSize	0.81	line	**1**	float
membership **start date**	date	date	startDate	1	startDate	1	date
participant **age**	age	age	participant	1	participant	1	integer
promotion_last_5years	promotion	binary	promotion	1	Nil	**0.00**	boolean
TotalWorking**Years**	years	numerical>=0	years	1	totalTime	0.77	integer
wt	weight	numerical>=0	weight	1	weight	1	float

Figure 5 – Semantic Type Assignment for FinalFormat and KGs

Overall, **FinalFormat** delivers finer-grained and more actionable classifications than ontology mappings or atomic datatypes, enhancing both semantic interpretability and data-quality profiling.

2.6 Feedback and Dictionary Enrichment

The system evolves through **manual analysis of NIL-defined** headers, cases where no matching format is found. After each run, these headers are aggregated, ranked by frequency, and reviewed to identify stable new or variant terms for the ***formats*** and ***abbreviations dictionaries***. This **human-in-the-loop** cycle converts NIL analysis into a structured feedback process, enabling continuous, evidence-based improvement of semantic type detection across diverse datasets.

2.7 Integration with Data Quality Assessment

As detailed in previous work [37], the framework connects semantic type assignments to automated data-quality rules. Each *FinalFormat* triggers specific quality checks (e.g., uniqueness, range, or format validation). The overall dataset quality is summarised by the ***HeadersIQ*** metric (defined in section 3.8), which aggregates all detected **Data Quality Issues (DQIs)** into a single interpretable score, supporting large-scale, automated profiling.

3 RESULTS AND COVERAGE ANALYSIS

This chapter presents a comprehensive evaluation of this header-centric semantic type detection framework across a diverse suite of tabular data benchmarks, encompassing both classic STI evaluation datasets and large, real-world sources. We compare our system's ***FinalFormat*** coverage and accuracy to state-of-the-art benchmarks and knowledge graph (KG) annotation systems (notably DBpedia [4] and Schema.org [20]), highlighting both quantitative performance and qualitative explainability through ***SourceKeywords***.

Scope clarification. All results in this section are obtained **from headers only**; no cell-content features are used. We therefore report **FinalFormat coverage** and **ontology mapping coverage** (DBpedia/Schema.org) under a header-only setting.

3.1 Datasets and Benchmarking Scope

We assess **scalability, robustness, and generalisability** over seven benchmarks (***Table 2***):

- **UCI** [24] – 50 classic datasets, relatively high columns-per-dataset, real-world domains.
- **Prague Relational Learning Repository** [26] – 50 curated relational tables.

- **Sato/VizNet** [45, 13] – large-scale web tables; **Sato** is a labelled subset of **VizNet**.
- **Kaggle** [18] – broad, real-world user-contributed tables spanning **17 areas**, heterogeneous headers.
- **T2Dv2** [35] – Wikipedia tables with **DBpedia** groundings (* denotes that the 39 "Areas" are DBpedia classes).
- **SOTAB** [20] – the WDC Schema.org Table Annotation Benchmark, with property/type tasks.

Table 2 – Structural Statistics of Benchmarks

Benchmark	Domains	Datasets	Columns	Columns per Dataset
UCI	6	50	922	18.4
Prague	9	50	520	10.4
Sato	N/A	2 254	4 587	2
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 (used below).

- **Universe A – All seven benchmarks.** Used in ***Tables 2, 3*** and ***6***.
- **Universe B – Kaggle + VizNet (118,639 columns).** Used in ***Tables 4*** **and** ***5*** to study distribution, long-tail behaviour, and **bounded numerical** coverage at scale.

3.2 Semantic Type Detection: Coverage and Diversity

What is measured.

- ***FinalFormat*** **Types**: number of distinct ***FinalFormats*** observed in each benchmark.
- ***SourceKeywords***: **unique, normalised header tokens** extracted across all tables (lexical diversity).
- ***DBpedia/Schema.org*** **Types**: number of **distinct ontology classes/properties** successfully linked from column headers to the respective ontologies after normalization and abbreviation expansion.

Findings.

Table 3 and ***Figure 5*** summarise results. **VizNet** and **Kaggle** show markedly higher ***SourceKeywords*** due to scale and heterogeneity, yet headers are still resolved into our **39 FinalFormats**, preserving concise meaning across many word variations.

Table 3: Coverage of FinalFormat, SourceKeywords, and KG Types

Benchmark	FinalFormat Types	Source Keywords	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

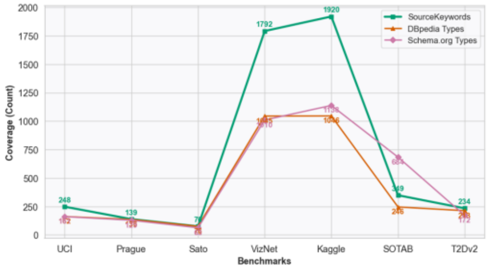


Figure 5: Coverage Comparison: SourceKeywords, KGs

Concretely, **only 15 core FinalFormats** (***Tables 4 and 5***) deterministically cover the *Sherlock*/*Sato* label space: **all 76 Sato formats map many-to-one into these 15 types**, and the two additional *Sherlock* labels (*File size* and *Team name*) do not appear in Sato's dataset. This consolidation preserves distinctions that matter for data quality (e.g., ***IDcolumn***, ***date***, ***postalcode***) while avoiding unnecessary granularity by merging near-duplicate labels (e.g., creator-like labels such as *Artist*, *Creator*, *Director*, *Jockey*, *Organisation*, *Publisher* → **name**; affiliation/brand-style variants → **categorical**).

Within **Universe B (K+V)**, these 15 core FinalFormats account for **≈80%** of all columns (***Tables 4*** **and** ***5***). The distribution is **skewed** towards a handful of operationally critical types, *categorical* (19.41%), *numerical≥0* (18.05%), *IDcolumn* (11.10%), *string* (9.50%), and *numerical* (7.64%), highlighting where robust handling most affects practice.

Table 4 – 15 *FinalFormats* that map all *Sherlock*/*Sato* Formats

#	*FinalFormats*	SherlockFormats	%
1	age	Age	1.04%
2	categorical	Affiliate, Affiliation, Album, Brand, Category, Class, Classification, Club, Code, Collection, Company, Continent, Education, Family, Format, Gender, Genre, Industry, Language, Location, Manufacturer, Origin, Position, Product, Ranking, Region, Religion, Result, Service, Sex, Species, Status, Symbol, Team, Type	19.41%
3	city	Birth place, City	0.85%
4	country	Country, Nationality	0.83%
5	date	Birth date	2.59%
6	day	Day	0.25%
7	IDcolumn	ISBN	11.10%
8	money	Currency	1.10%
9	name	Artist, Creator, Director, Jockey, Name, Operator, Organisation, Owner, Person, Publisher, Team name	5.45%
10	numerical	Credit, Elevation, Range	7.64%
11	numerical>=0	Area, Capacity, Depth, Duration, File size, Grades, Rank, Sales, Weight	18.05%
12	state	State	0.50%
13	street	Address	0.28%
14	string	Command, Component, Description, Notes, Order, Plays, Requirement	9.50%
15	year	Year	1.72%
		SUB TOTAL	**80.31%**

Beyond *Sherlock*/*Sato*, we covered an additional **23 formats** (***Table 5***), notably **bounded numericals** (e.g., *percentage*, chemical related (*ph*, *acidity*, *alkalinity*), geographically related (*latitude/longitude*), health related (*heartrate*, *bloodpressure*)), enabling finer-grained DQI checks that traditional taxonomies omit. These add up to around **6%, totalling over 7,000 columns in this universe, which is quite considerable.**

Table 5 - Additional Formats not considered by *Sherlock*/*Sato*

#	FinalFormat	%	#	FinalFormat	%
1	binary	1.53	14	datetime	0.17
2	acidity	0.03	15	E-mailformat	0.13
3	alkalinity	0.00	16	IPformat	0.12
4	angle	0.03	17	modelname	0.01
5	bloodpressure	0.11	18	month	0.14
6	heartrate	0.18		**NIL**	**13.39**
7	hour	0.04	19	phone	0.17
8	latitude	0.14	20	postalcode	0.10
9	longitude	0.14	21	time	0.85
10	normalized	0.05	22	URLformat	0.34
11	percentage	1.89	23	week	0.07
12	ph	0.02	24	weekday	0.05
13	saltness	0.00		TOTAL	19.69

In **Universe B**, **NIL = 13.39%** (thus **Valid = 86.61% = ~87%**) over **118,639** headers. Here, **NIL** denotes columns with **no match** after normalisation and abbreviation expansion. This rate is low given the corpus size and heterogeneity, and it quantifies the remaining long-tail of noisy or novel headers.

Scalability evidence. Relative to the initial UCI phase (~1,000 columns [37]) where a dictionary of 1,800 format terms + 300 abbreviations (~2,100 entries) achieved local coverage, a **modest ~80 dictionary expansion** to **2,800 + 1,000** entries enabled generalisation to Kaggle+VizNet with **≈103,000 valid assignments (≈87)**, a **≈103× increase in practical coverage capacity**.

Table 6 - Scalability evidence

Phase	Dictionary size	Columns evaluated	Valid assigned
UCI baseline	1,800 + 300 ≈ 2,100	~1,000	~1,000 (near-perfect)
Kaggle+VizNet	2,800 + 1,000 ≈ 3,800	118,639	~103,000 (≈87%)

Ongoing analysis of these ≈103,000 annotated columns shows that roughly **half** are direct, unambiguous matches between column headers and dictionary terms, while the remaining **≈ 51,500 multi-word columns** require statistical validation to confirm correctness, forming the basis of our current research direction.

Bounded numericals. ***Appendix B*** [38] lists all the adopted limits for certain numeric formats (e.g., *age* 0–130, *ph* 0–14, *latitude* −90–90). These limits indicate natural ranges that can be validated; when values fall outside them, corresponding Data Quality Issues (DQIs) are reported [37].

3.3 NIL Token improvement process.

A qualitative review of the top 100 **NIL** cases in Universe B, reveals that most unresolved cases are extremely short codes or symbols (e.g., *no, so, ga, sv, gf, cg, sf, dp, pp, po*), which carry little semantic information and therefore fall outside any meaningful format mapping. A smaller subset toward the end of the list (e.g., *streak, medal, designation, vote, stats, defense, winnings, subcategory, msrp*) shows domain-specific or abbreviative potential that may justify later dictionary extension.

These observations confirm that the current NIL proportion corresponds mainly to low-information or highly domain-specific headers rather than systemic dictionary gaps. The **NIL Token Improvement Process** operates as a controlled feedback loop: ranking unmatched tokens, inspecting them, and selectively enriching dictionaries only for stable, interpretable terms. This ensures continuous yet disciplined coverage growth across benchmark iterations. The complete ranked list of the top 100 tokens appears in ***Appendix C*** [38].

3.4 Validity, NIL Rates, and Robustness

Table 7 compares **NIL counts** and **Valid** for (i) our **FinalFormat** assignments and (ii) subsequent mappings to **DBpedia** and **Schema.org** across **Universe A**.

Table 7 - NIL/Valid Assignment Rates

Bench mark	Final Format NIL	Final Form at Valid %	DB pedia NIL	DB pedia Valid %	Schema .org NIL	Schema. org Valid %
UCI	12	98.7	433	53	413	55.2
Prague	0	100	94	81.9	93	82.1
Sato	0	100	0	100	423	90.8
VizNet	9 435	87.4	21 244	71.6	23 795	68.2
Kaggle	7 274	85.4	16 546	66.7	17 467	64.9
SOTAB	5	99.3	163	77.9	7	99.1
T2Dv2	267	77.2	305	74	469	60

Definition of Valid and NIL.

For all systems, *Valid* = (1 − NIL / Total Columns) × 100. For FinalFormat, a column is considered *valid* when its header is successfully mapped to one of the 39 semantic types after all dictionary and abbreviation expansions. For DBpedia and Schema.org, *Valid* indicates the fraction of columns whose FinalFormat label could be semantically aligned with at least one ontology property or class, measuring mapping coverage.

NIL *= no dictionary match after all header normalisation and abbreviation/variant expansions..*

Key Observations:

- ***FinalFormat*** consistently attains the highest Valid , as it represents the base classification before ontology mapping.
- Mapping to DBpedia and Schema.org remains robust, with most FinalFormat types finding at least one compatible ontology concept.
- **Sato** and **Prague** achieve **100 Valid assignments**, confirming the **completeness** of our dictionaries and rules on structured benchmarks while also showing that this framework **fully subsumes** the *Sherlock* label space.
- **UCI** and **Kaggle** exhibit the expected drop in KG mapping validity due to diverse, noisy headers and domain-specific terms absent from current ontologies.

3.5 Machine Learning Semantic Type Detection System

We evaluated five classifiers (Random Forest, Gradient Boosting, K-Nearest Neighbors, Logistic Regression, and Linear SVC) using the preprocessing and balancing strategy described in ***Section 2.4***. Class imbalance was particularly relevant for minority semantic types (e.g., *URLformat*, *Time*), where we applied SMOTE to improve recall. Models were tuned with GridSearchCV and assessed by precision, recall, and F1-score on an 80–20 split. **The data sources used were the 50 UCI datasets and the 50 Prague Relational Data Tables.**

Figures 6-8 present per-format results. Categorical columns showed the most variability across models, while the distinction between *Numerical* and *Numerical>=0* remained a frequent source of confusion. Gradient Boosting performed worst on ambiguous or NIL cases, whereas Logistic Regression and LinearSVC were more robust.

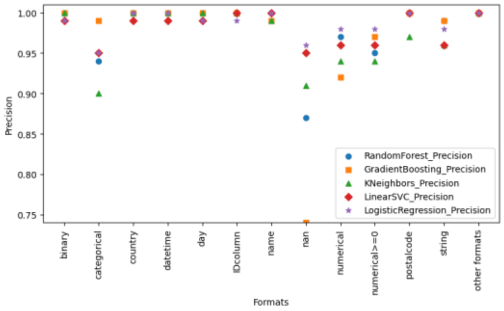


Figure 6 - Precision for Formats on 5 ML Models

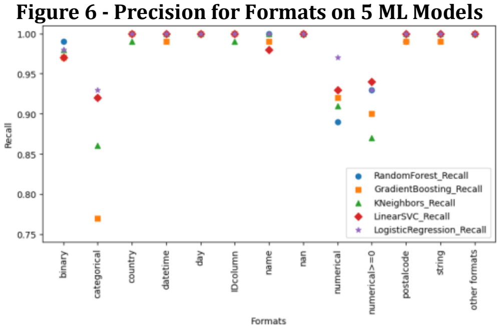


Figure 7 - Recall for Formats on 5 ML Models

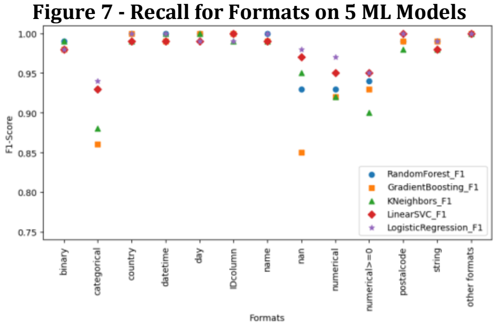


Figure 8 - F1-Score for Formats on 5 ML Models

Several formats consistently achieved perfect precision, recall, and F1-score across all classifiers, including *Email*, *URL*, *Age*, *City*, *Country*, *Date*, and *Phone*. ***Figure 9*** summarises average performance across the five models.

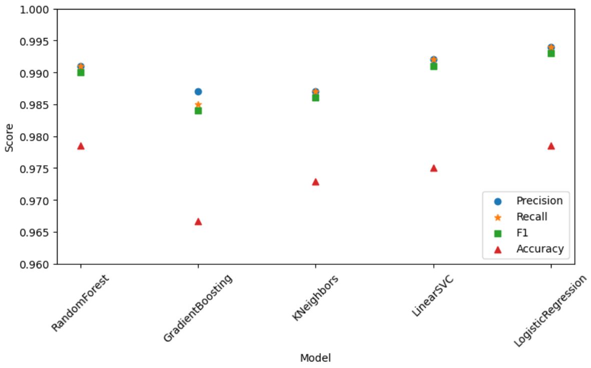


Figure 9 - Average Metrics on 5 ML Models

The near-perfect results observed in ***Figures 6-9*** arise from the controlled scope of the machine-learning subset rather than overfitting. The evaluation covered only about **1,400 columns**, a scale small enough to allow **full manual verification of the training and test labels**, ensuring almost noise-free supervision. This explains why several semantic types (e.g., *Email*, *URL*, *Date*, *City*) achieved F1 ≈ 1 across all classifiers.

In contrast, the **large-scale rule-based analysis**, spanning more than **118,000 columns** that were not processed by the ML models, revealed around **13% NIL cases**, reflecting unresolved or ambiguous types under real-world variability. Consequently, the ML results should be interpreted as an **upper-bound performance under perfect annotation**, while the large-scale NIL proportion represents the **lower-bound realism** of open-domain data. This distinction clarifies that the high metrics result from dataset certainty, not optimistic bias or information leakage.

3.6 Handling Complex and Ambiguous Header Cases

This system is designed to tackle a wide variety of challenging header formats and ambiguous lexical cues through sophisticated normalization and keyword extraction, ensuring precise semantic type assignment even in noisy or unconventional scenarios.

Table 8 - Detailed FinalFormat and SourceKeywords allocation

Original Column	**SourceKeywords**	**FinalFormat**
Members with **age** 5 – 17 years old	**age**	**age**
IsBasedOnRealStory	**is**	**binary**
BookedHotel**OrNot**	**or not**	**binary**
trestbps	**blood pressure**	**bloodpressure**
ATTEND_**DEPT**	**department**	**categorical**
Date Range	**range**	**categorical**
Father's **Birth Place**	**birth place**	**city**
dtRef	**date**	**date**
Request **timestamp**	**timestamp**	**datetime**

Table 8 provides a list of examples for the allocation of **SourceKeywords** and **FinalFormats** from Universe 2(K+V).

For example, **binary-type** columns are often signaled by common words such as "**is**", or terms like "**or not**", which trigger classification into the **binary** FinalFormat. This allows for accurate detection of boolean columns besides explicit true/false values.

Columns containing the word ***“range”*** are correctly identified as categorical despite co-occurring terms like ***“date”*** or “*price*” that might otherwise mislead the classifier towards **temporal** or monetary types. This demonstrates the system’s nuanced **contextual understanding,** supported by dictionary-based semantic disambiguation.

Geographic and demographic terms also benefit from semantic synonym mapping. Headers containing “***birth place***” are consistently mapped to the ***city*** FinalFormat. Similarly, “***dt***” fields are recognized as date types, and “***timestamp***” keywords indicate ***datetime*** types.A novel addition is the recognition of medical terms such as “*trest**bps***”, representing blood pressure measurements, which are accurately classified under the ***bloodpressure*** FinalFormat, demonstrating the system’s adaptability to specialized domains.

Through this comprehensive keyword and abbreviation-driven approach, the system maintains high precision in semantic type detection, enabling actionable data quality rules that rely on correctly interpreted column semantics. This is especially critical for heterogeneous, real-world datasets where inconsistent or abbreviated headers are commonplace. ***Appendix D*** [38] describes in detail the rule-based preprocessing, normalization, and abbreviation-driven mapping steps that precede the optional machine-learning classifier presented in ***Appendix A***.

3.7 Comprehensive Benchmark Results on the SemTab 2024 Metadata-to-KG Track

Ground-truth note. As detailed in ***Appendix E***, the SemTab-2024 Metadata-to-KG ground truth (GT) exhibits typical large-benchmark issues (duplicates, label-granularity mismatches, occasional mislinks). We therefore report **official GT-strict** results for head-to-head comparison and, separately, **blinded header-only diagnostics** for our outputs.

Official (GT-strict) results. Using the organizers’ ground truth and script, our system correctly annotated **64/141** columns (**Hit@1 = 0.45**, 95% CI ≈ **[0.37, 0.54]**) and **68/141** (**Hit@5 = 0.48**, 95% CI ≈ **[0.40, 0.56]**). Baselines for Adwan and CVA/MetaLinker are taken from the SemTab-2024 official report [12]. **Panel A of *Table 9*** lists these head-to-head figures.

Internal, header-only diagnostic audits (ours only). We conducted a **blinded, post-hoc** audit of our predictions (***Appendix E***). The author, **blinded to official GT labels and to all cell values**, adjudicated each output using a pre-specified rubric: **C.1 GT-Confirmed** (same canonical entity), **C.2 GT-Refinement** (author-proposed replacement: ontology-consistent and more appropriate for the header), **C.3 Ontology-Consistent Alternative** (plausible but not GT), and **C.4 Incorrect**. **Panel B of Table 9** reports cumulative diagnostics (**B1 strict+C.1, B2 +C.2, B3 +C.3**) and 95% CIs. These diagnostics quantify **GT noise** and header-only ambiguity; they are **not** used for cross-system ranking.

Reading Panel B. In our audit excerpt, **C.2 GT-Refinement** contributes **43/141 (30.5%)** and **C.3 Alternatives** contribute **6/141 (4%)**. Thus, diagnostic Hit@1 rises from **0.45** (strict) to **≈0.76** when adding E.2, and to **≈0.80** when adding E.3 (95% CIs in ***Table 9/Appendix E***). This suggests many residual “errors” under strict GT stem from **label granularity/aliasing** rather than model mismatch, particularly in **header-only** settings.

Metric note. We also report **NIL rate** (no match after all expansions); when shown, **Valid headers = 1 − NIL**. **Hit@k** follows the standard definition irrespective of NIL.

Cross-system conclusions rely only on Panel A (GT-strict). Panel B is a **diagnostic** lens for GT quality and practical metadata semantics.

Table 9 - Comparative Analysis/SemTab 2024 Metadata track

Panel A - Official (GT-strict)

System	Hit@1	Hit@5
Ours (GT-strict)	0.45 [0.37, 0.54]	0.48 [0.40, 0.56]
Adwan [43]	0.75	0.92
CVA / MetaLinker [25]	0.55	0.70

Panel B - Diagnostic (ours only; header-only audit)

Category	Hit@1	Hit@5
B1 – Strict + C.1 (GT-Confirmed)	0.45 [0.37, 0.54]	0.48 [0.40, 0.56]
B2 – B1 + C.2 (GT-Refinement)	0.759 [0.68, 0.82]	0.79 [0.71, 0.85]
B3 – B2 + C.3 (Alternative)	0.801 [0.73, 0.86]	0.83 [0.76, 0.88]

3.8 *HeadersIQ*: A Metric for Data/Information Quality Monitoring

We introduce ***HeadersIQ***, a simple yet effective metric that quantifies the overall data/information quality of a data source by reflecting the proportion of clean data items it contains.

The name ***HeadersIQ*** reflects its foundation in column **headers** and its focus on **Information Quality (IQ)**, while also evoking the idea of intelligent assessment, inspired by the concept of **Intelligence Quotient**.

Formal Definition.

Let:

N_c = number of columns,

N_r = number of rows,

$T = N_c \times N_r$ = total number of data items (cells),

D = {d1,d2,...,dk} = set of all **Data Quality Issues (DQIs)** evaluated (e.g., missing values, type mismatch, range violation, duplication, format inconsistency, etc.),

E_i = number of violations detected for DQI d_i.

Then the total number of detected issues is:

$$E = \sum_{i=1}^{k} E_i$$

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

$$\left(\mathbf{1} - \frac{\boldsymbol{E}}{\boldsymbol{T}}\right) \times \mathbf{100}$$

Normalization and Interpretation.

Because both $\boldsymbol{E}$ and $\boldsymbol{T}$ are measured at the cell level, the ratio $\boldsymbol{E}/\boldsymbol{T}$ inherently normalizes by dataset size, allowing direct comparison across datasets.

A score of 100 indicates a fully clean dataset ($\boldsymbol{E}$ = 0), lower values represent the proportion of data items that violated the expected rules or formats.

Example:

For dataset 45 (303 rows, 14 columns → $\boldsymbol{T}$=4242) and $\boldsymbol{E}$=7 total Data Quality Issues (Errors):

HeadersIQ = (1 – 7/4242) * 100 ≈ 99.84

This indicates that 99.84 of all values in the dataset are clean, based on the identified issues.

Table 10 shows how *HeadersIQ* varies across the first ten datasets from UCI (including dataset **45**), with values ranging from 93.14 to 100, offering a quick overview of cleanliness levels.

Table 10 - *HeadersIQ* values for the first ten datasets from UCI

Dataset	Total Rows	Columns	Total Data Items	Total DQIs – Errors	HeadersIQ
52	150	5	750	0	100
45	303	14	4 242	7	99.84
2	32 561	15	488 415	4 262	99.13
107	178	14	2 492	170	93.18
42	214	11	2 354	0	100
58	20 000	17	340 000	0	100
142	10 000	21	210 000	1 000	95.24
92	4 601	58	266 858	0	100
103	435	17	7 395	392	94.7
27	690	16	11 040	757	93.14

HeadersIQ reflects the proportion of clean data items in each dataset, computed as $\left(1-\frac{\boldsymbol{E}}{\boldsymbol{T}}\right)\times 100$, where $\boldsymbol{E}$ is the total detected issues and T the total data items. Implemented in the notebook *Attribute-BasedDataQualityAssessment.ipynb*[38].

This metric is especially valuable in real-world scenarios where hundreds of datasets may be evaluated over time. *HeadersIQ* supports scalable, automated data quality monitoring across data lakes and pipelines, providing clear signals for data governance teams and reducing manual inspection effort.

These results also demonstrate that *HeadersIQ* provides a dataset-level quality signal consistent with the **SemTab 2024 "*IsGold?*"** objective of assessing the overall reliability by automatically identifying structural noise, annotation inconsistencies, and NIL misuse.

4 DISCUSSION

This chapter discusses broader implications, revisits core design decisions, and assesses contributions. It synthesizes insights across prior chapters to articulate the impact and future potential of this explainable, header-centric framework for semantic table interpretation and data quality assessment.

4.1 Reevaluating the Value of Header-Centric Type Detection

Historically, header-centric annotation was viewed as unreliable, with Sherlock [14] and Sato [45] promoting value-driven methods. Our findings counter this: with systematic preprocessing, a dynamically updated ***formats dictionary***, and an optional ML classifier, column headers act as rich semantic indicators, enabling precise annotation and scalable profiling. ***FinalFormat***'s 39 interpretable types bridge overly broad atomic types and granular ontology classes, supporting both human interpretability and actionable data quality rules.

4.2 Scalability and NIL Token Improvement Process

In this work we decided to analyse separately a superset of **118,639** columns from **Kaggle + VizNet**, providing a heterogeneous stress-test for header-only assignment and NIL analysis (see Section 3.2). A modest **~80** dictionary expansion enabled a **≈103×** increase in practical coverage capacity, evidencing strong **reusability** and **scalability of the header-centric approach**. About half of the valid assignments are direct dictionary matches; the rest are **multi-word headers** now informing targeted enrichment.

NIL Token Improvement Process: After each large-scale run, unmatched headers (NIL) are aggregated, frequency-ranked, and reviewed to separate low-information tokens from genuinely missing types; only the latter trigger dictionary/abbreviation expansion. This disciplined loop keeps growth interpretable and sustainable, extending scalability from processing capacity to **semantic** scalability.

4.3 Comparative Evaluation with the *Sato/Sherlock* Benchmark

FinalFormat generalised all 76 Sato formats into just 15 categories and added 23 types (chemistry, geolocation, health). These new types account for ~6% of the 118,639 columns, showing that a compact, human-readable system captures broad domain diversity without excessive granularity. The Sherlock/Sato taxonomy is fully contained within ***FinalFormat***, while the added types extend coverage beyond those benchmarks to larger, noisier repositories.

4.4 Broader Benchmark Evaluation and Ontology Mapping

We evaluated our semantic assignments against established knowledge graphs (DBpedia and Schema.org) and across other widely recognized benchmarks, besides Kaggle and Viznet, including UCI [24], Prague [26], Sato [45], SOTAB [20], and T2Dv2 [35]. These comparisons showed:

- High alignment between our predicted types and KG properties, demonstrating semantic compatibility and interpretability.
- Broader applicability of our FinalFormats, which could often match or outperform ontology terms in real-world coverage.
- Robust results even for schema-poor sources like Kaggle, where header variability is high and metadata sparse.

These results highlight the practical strengths of ***FinalFormat*** in bridging the gap between ontology-based and statistical annotation approaches.

4.5 Machine Learning Performance and Format-Specific Insights

The optional ML classifier performed strongly on well-defined types and ambiguity mitigation. Categorical types showed model variability (e.g., Gradient Boosting: Recall 0.77, F1 0.86). Distinguishing Numerical vs Numerical≥0 remained subtle (F1 0.89–0.98). **NIL** detection varied by model (Logistic Regression F1 0.98; Linear SVC F1 0.97; Gradient Boosting Precision 0.74).

Interpreting ML metrics.

The high scores come from a small, clean subset (∼1,400 columns – 50 datasets from UCI + 50 tables from Prague) with verified labels (best-case performance). At corpus scale (∼118k headers, not run through the classifier), ∼13% are **NIL**, reflecting real-world ambiguity; use that as the practical lower bound.

4.6 *HeadersIQ*: A Simple Yet Powerful Metric

As detailed in Section 3.7, *HeadersIQ* proved both interpretable and stable, suitable for large-scale data quality monitoring and consistent with the **SemTab 2024 "IsGold?"** objective of assessing dataset-level reliability.

4.7 Rethinking Ground Truth and Benchmark Comparability

SemTab-2024 Metadata revealed ground-truth discrepancies; many "errors" were valid alternates under human review. Using the organizers' GT we obtain **Hit@1/5 = 0.45/0.48**; after reconciliation (***Appendix E***) **0.80/0.83**; thus accuracy must be read with **semantic appropriateness**, and rigid gold labels can underestimate practical correctness. Future STI/CTA benchmarks should support **graded or multi-label** correctness with **human-in-the-loop** verification.

4.8 Limitations and Remaining Challenges

- **Metadata dependency**. The pipeline assumes informative headers; cryptic/sparse labels reduce semantic assignment accuracy and downstream DQ assessment.
- **Residual multi-word/NIL cases**. Despite dictionary growth and the optional classifier, a long tail of multi-word headers remains under review; coverage depends on the ongoing enrichment loop.
- **Ontology mapping gaps**. Some FinalFormats have weak or ambiguous alignments in DBpedia/Schema.org, limiting property/class mappings in noisy domains.
- **Evaluation scope**. Classifier metrics reflect a small, clean subset; large-scale results (header-only) capture open-domain variability. A 95%-confidence audit of multi-word assignments (with humans + LLM triage) is planned to quantify true correctness.

4.9 Future Directions

- **LLM integration for enrichment & disambiguation:** Use transformer models (e.g., BERT/GPT) to interpret ambiguous headers, propose candidate synonyms/ abbreviations and **FinalFormat** labels and assist the **optional classifier** when confidence is low.
- **Statistical audit** with humans + LLMs (95% confidence): Run a 95%-confidence audit of the ≈51.5k multi-word assignments (target margin of error to be defined), using LLM triage to rank candidates and independent human checks (double-annotation or small majority). Report %Correct / %Incorrect / %Ambiguous→NIL, update dictionaries/rules accordingly, and re-evaluate the affected slice to track ΔValid%.
- **Multilingual expansion**: Extend formats/abbreviations to Portuguese first, leveraging the same LLM-assisted enrichment + human verification loop.

4.10 Broader Impacts and Practical Adoption

The framework applies to data integration, metadata management, ETL automation, and KG construction. Its explainability, scalability, and adaptability align with trustworthy data needs. Integrating semantic understanding and data quality assessment in one pipeline sets a precedent for future STI systems. By uniting dynamic dictionaries, robust header analysis, explainable type assignments, and actionable quality metrics, it lays the foundation for intelligent metadata-driven governance in large-scale, heterogeneous environments.

5 RELATED WORK

This section reviews prior research in **Semantic Table Interpretation (STI)** and **Column Type Annotation (CTA),** situating this work within current trends and highlighting gaps that motivate our design. We follow the structured approach of **Liu et al. [23],** who proposed a taxonomy for categorizing CTA systems based on annotation scope, metadata usage, reliance on headers, and knowledge graph integration.

5.1 Semantic Table Interpretation and CTA Tasks

STI aims to make tabular data semantically meaningful by linking table structures to knowledge-graph entities and types. Among its subtasks, **Column Type Annotation (CTA)** is central to this work.

Recent **SemTab challenges** (e.g., 2024 Metadata-to-KG) introduced **metadata-only** tracks that require systems to infer semantic types without cell values, motivating scalable and interpretable **header-centric** methods.

5.2 Column Type Annotation Paradigms

Based on taxonomy proposed by **Liu et al. [23]**'s **Table 1**, CTA systems can be grouped into four broad families:

- **Heuristic and Lookup-Based Systems**: Use string matching between headers and dictionary entries or KG labels. Examples include Wang et al [44], C^2 [19], MAGIC [40], and Alobaid et al [3].

- **Heuristic and Iterative Systems**: Apply rule refinement or contextual propagation across table elements, such as TableMiner+ [47], CSV2KG [41], T2K [34] and MTab [29, 30, 31].
- **Feature-Based (Shallow ML) Systems**: Incorporate engineered features like token overlap, column length, and label similarity. Limaye et al. [22] and Mulwad et al. [27, 28] fall into this group.
- **Deep Learning-Based Systems**: Leverage embeddings or transformers trained on table corpora. Examples include *Sherlock* [14], *Sato* [45], *TURL* [10], *Doduo* [42], *RECA* [39] and *DAGOBAH* [6].

Only a **small subset** of 12 systems among the 30 exhibited rely exclusively on header-level features, and even fewer operate without cell values. This gap motivates our metadata-only, header-centric framework, whose extended comparison is summarized in ***Table 11*** and ***Appendix F***.

5.3 Comparative Gaps and Trends

To address this gap, we extend Liu et al. [23]'s Table 1 with recent systems (including LLM-based metadata methods), a header-only flag, and SemTab track notes; see ***Appendix F*** [38] for the full, updated ***Table F*** and curation criteria. A condensed extract is shown below.

Table 11 - Condensed extract from the full comparative analysis (*Table F* in *Appendix F* [38]), adapted and expanded from Table 1 in Liu et al. [23]

System (year)	Header-only?	KG	SemTab note
Wang et al. (2012)	No	Probase	Baseline
Sherlock (2019)	No	DBpedia	Baseline
Sato (2020)	No	DBpedia	Baseline
ADWAN (2024)	**Yes**	DBpedia, Schema.org	2024 Metadata-only **Winner**
CVA/MetaLinker (2024)	**Yes**	DBpedia, Schema.org	2024 Metadata-only Participant
[37] (2024)	**Yes**	–	UCI study
This paper (2025)	**Yes**	DBpedia, Schema.org	2024 Metadata-only **Offline comparison**

A few trends emerge. Early and deep table models, Wang et al. (2012), Sherlock (2019), Sato (2020), were **not header-only**, despite strong KG use. The 2024 metadata-only track re-centered headers and produced competitive **header-only** systems (ADWAN winner; CVA/MetaLinker participant) using LLM-driven methods with **DBpedia + Schema.org**.

Prior study (**[37], 2024**) was **header-only** on UCI (no KG mapping). **This paper (2025)** is likewise **header-only** with dual-KG mapping (DBpedia + Schema.org) and has been compared in this study with Semtab 2024 metadata-only.

Overall, table 11 highlights (i) the scarcity of genuine **header-only** approaches before 2024, (ii) the rise of **LLM-assisted metadata** methods, and (iii) this work's distinct combination of explainable, header-centric typing, dynamic dictionary enrichment (***SourceKeywords***), dual-KG mapping, and dataset-level data quality signals (***HeadersIQ***).

5.4 Key Lessons from Liu et al.

Liu et al. [23] surface several insights that directly influenced our system design:

- **Metadata as a Primary Signal**: While most systems focus on cell content, Liu's taxonomy highlights that metadata such as headers are rich yet underexploited. This insight guided this header-centric design, ensuring scalability even in metadata-only scenarios.
- **Benchmark Bias Toward Single Ground Truth**: Existing benchmarks often penalize cases where multiple semantic types could be valid. This approach accommodates ambiguity via **NIL** and report human-reconciled analyses
- **Knowledge Graph Sparsity**: KGs frequently lack adequate coverage for noisy or domain-specific headers. We introduced **FinalFormat** as an interpretable intermediate layer that bridges headers and **DBpedia/Schema.org**.

By aligning with Liu's framework and situating our contribution within this extended comparative analysis (***Table 11*** and ***Table F*** in ***Appendix F***), we contribute to a clearer understanding of trade-offs and challenges in modern CTA research.

6 CONCLUSION

We presented an explainable, header-centric framework for semantic type detection and data quality assessment. Unlike approaches that rely on cell values or deep models trained on static taxonomies, our method uses curated, feedback-enriched dictionaries plus an optional ML classifier to infer types directly from headers, enabling metadata-only operation (as in SemTab-2024). With only ~80% dictionary growth, the framework achieved ~87% header-only assignments across >118k columns and introduced ***HeadersIQ***, a simple, formally defined dataset-level quality metric.

Our evaluation spans classic and recent benchmarks: the Sato/Sherlock label space (76 types) is fully contained within just 15 ***FinalFormats***, and a SemTab-2024 Metadata-to-KG offline comparison contextualises results alongside ADWAN and CVA/MetaLinker. Overall, the system delivers competitive performance with broader coverage, dual-KG mapping (DBpedia/Schema.org), and token-level traceability via ***SourceKeywords***.

We also **extend Liu et al.'s comparative taxonomy** with an updated matrix (***Appendix F***), covering recent systems and metadata-only criteria.

Future work: run a 95%-confidence audit of multi-word assignments using LLM triage plus independent human checks to drive dictionary/rule updates; use LLMs to propose synonyms/abbreviations and assist the optional classifier on low-confidence cases; and extend multilingual coverage via the same enrichment loop.

REFERENCES

[1] Nora Abdelmageed. 2024. SemTab 2024 – IsGold? Track. SemTab 2024 challenge website. [Online]. Available: https://sem-tab-challenge.github.io/2024/tracks/is-gold-track.html

[2] Nora Abdelmageed; Ernesto Jiménez-Ruiz; Oktie Hassanzadeh; Birgitta König-Ries. 2025. KG2Tables: A Domain-Specific Tabular Data Generator to Evaluate Semantic Table Interpretation Systems. Transactions on Graph Data and Knowledge 3, 1 (Apr. 2025), Article 1, 1–28. https://drops.dagstuhl.de/storage/08tgdk/tgdk-vol003/tgdk-vol003-issue001/TGDK.3.1.1/TGDK.3.1.1.pdf

[3] Ahmad Alobaid; Oscar Corcho. 2022. Balancing coverage and specificity for semantic labelling of subject columns. Knowledge-Based Systems 240 (2022), Article 108092. https://doi.org/10.1016/j.knosys.2021.108092

[4] Sören Auer; Christian Bizer; Georgi Kobilarov; Jens Lehmann; Richard Cyganiak; Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In The Semantic Web – ISWC 2007/ASWC 2007, LNCS 4825, Springer, 722–735. https://www.cis.upenn.edu/~zives/research/dbpedia.pdf

[5] Michael J. Cafarella; Alon Y. Halevy; Daisy Zhe Wang; Eugene Wu; Yang Zhang. 2008. WebTables: Exploring the power of tables on the web. Proceedings of the VLDB Endowment 1, 1 (2008), 538–549. https://yz.mit.edu/old-site/papers/webtables-vldb08.pdf

[6] Yoan Chabot; Thomas Labbé; Jixiong Liu; Raphaël Troncy. 2019. 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, 41–48. https://www.cs.ox.ac.uk/isg/challenges/sem-tab/2019/papers/DAGOBAH.pdf

[7] Marco Cremaschi; Fabio D'Adda; Fidel Jiomekong Azanzi; Jean Petit Yvelos; Ernesto Jiménez-Ruiz; Oktie Hassanzadeh. 2025. SemTab 2025: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching. SemTab 2025 challenge website. https://sem-tab-challenge.github.io/2025/

[8] Vincenzo Cutrona; Jiaoyan Chen; Vasilis Efthymiou; Oktie Hassanzadeh; Ernesto Jiménez-Ruiz; Juan Sequeda; Kavitha Srinivas; Nora Abdelmageed; Madelon Hulsebos; Daniela Oliveira; Catia Pesquita. 2021. Results of SemTab 2021. CEUR Workshop Proceedings 3103, 1–12. https://ceur-ws.org/Vol-3103/paper0.pdf

[9] DBheaders file. 2014. WebTables (University of Mannheim). [Online]. Available: data.dws.informatik.uni-mannheim.de/webtables/2014-02/statistics/DBheaders.txt

[10] Xiang Deng; Huan Sun; Alyssa Lees; You Wu; Cong Yu. 2021. TURL: Table Understanding through Representation Learning. Proceedings of the VLDB Endowment 14, 3 (Oct. 2021), 307–319. https://www.vldb.org/pvldb/vol14/p307-deng.pdf

[11] Ivan Ermilov; Axel-Cyrille Ngonga Ngomo. 2016. TAIPAN: Automatic Property Mapping for Tabular Data. In Knowledge Engineering and Knowledge Management: EKAW 2016, LNCS 10024, Springer, Cham, 163–179. https://doi.org/10.1007/978-3-319-49004-5_11

[12] Oktie Hassanzadeh; Nora Abdelmageed; Marco Cremaschi; Vincenzo Cutrona; Fabio D'Adda; Vasilis Efthymiou; Benno Kruit; Elita Lobo; Nandana Mihindukulasooriya; Nhan H. Pham. 2024. Results of SemTab 2024. CEUR Workshop Proceedings 3889, 1–11. https://ceur-ws.org/Vol-3889/paper0.pdf

[13] Kevin Z. Hu; Neil S. Gaikwad; Michiel A. Bakker; Madelon Hulsebos; Emanuel Zgraggen; César A. Hidalgo; Tim Kraska; Guoliang Li; Arvind Satyanarayan; Çağatay Demiralp. 2019. VizNet: Towards a large-scale visualization learning and benchmarking repository. In Proceedings of CHI 2019. https://dl.acm.org/doi/10.1145/3290605.3300892

[14] Madelon Hulsebos; Kevin Z. Hu; Michiel A. Bakker; Emanuel Zgraggen; Arvind Satyanarayan; Tim Kraska; Çağatay Demiralp; César A. Hidalgo. 2019. Sherlock: A deep learning approach to semantic data type detection. In Proceedings of ACM SIGKDD 2019, 1500–1508. https://dl.acm.org/doi/10.1145/3292500.3330993

[15] Madelon Hulsebos; Paul Groth; Çağatay Demiralp. 2023. AdaTyper: Adaptive Semantic Column Type Detection. https://arxiv.org/abs/2311.13806

[16] Ernesto Jiménez-Ruiz; Oktie Hassanzadeh; Vasilis Efthymiou; Jiaoyan Chen; Kavitha Srinivas. 2020. SemTab 2019: Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems. In Proceedings of the 17th International Semantic Web Conference (ESWC 2020), 514–530. https://link.springer.com/chapter/10.1007/978-3-030-49461-2_30

[17] Ernesto Jiménez-Ruiz; Oktie Hassanzadeh; Vasilis Efthymiou; Jiaoyan Chen; Kavitha Srinivas; Vincenzo Cutrona. 2020. Results of SemTab 2020. CEUR Workshop Proceedings 2775, 1–8. https://ceur-ws.org/Vol-2775/paper0.pdf

[18] Kaggle Datasets. Kaggle (website). Available: https://www.kaggle.com/datasets

[19] Udayan Khurana; Sainyam Galhotra. 2020. Semantic Annotation for Tabular Data. https://arxiv.org/abs/2012.08594

[20] Keti Korini; Ralph Peeters; Christian Bizer. 2022. SOTAB: The WDC Schema.org Table Annotation Benchmark. In Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2022), CEUR Workshop Proceedings. https://ceur-ws.org/Vol-3320/paper1.pdf

[21] Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. 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). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 75–76. https://doi.org/10.1145/2872518.2889386

[22] Girija Limaye; Sunita Sarawagi; Soumen Chakrabarti. 2010. Annotating and searching web tables using entities, types and relationships. Proceedings of the VLDB Endowment 3, 1–2 (2010), 1338–1347. https://dl.acm.org/doi/10.14778/1920841.1921005

[23] Jixiong Liu; Yoan Chabot; Raphaël Troncy; Viet-Phi Huynh; Thomas Labbé; Pierre Monnin. 2023. From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. Journal of Web Semantics 76 (2023), Article 100761. https://doi.org/10.1016/j.websem.2022.100761

[24] Markelle Kelly; Rachel Longjohn; Kolby Nottingham. 2025. The UCI Machine Learning Repository. University of California, Irvine. Available: https://archive.ics.uci.edu

[25] Margherita Martorana; Xueli Pan; Benno Kruit; Tobias Kuhn; Jacco van Ossenbruggen. 2024. Column Vocabulary Association (CVA): Semantic Interpretation of Dataless Tables. In Proceedings of SemTab 2024, CEUR Workshop Proceedings. https://sem-tab-challenge.github.io/2024/semtab2024-proceedings/paper2.pdf

[26] Jan Motl; Oliver Schulte. 2015. The Prague Relational Learning Repository. In Proceedings of the 25th International Conference on Inductive Logic Programming (ILP 2015), LNCS 9616, Springer, 93–99. https://arxiv.org/html/1511.03086v2

[27] Varish Mulwad; Tim Finin; Anupam Joshi. 2013. Semantic message passing for generating linked data from tables. In Proceedings of ISWC 2013, LNCS 8218, Springer, 311–326. https://dl.acm.org/doi/10.1007/978-3-642-41335-3_23

[28] Varish Mulwad; Tim Finin; Zareen Syed; Anupam Joshi. 2010. Using linked data to interpret tables. In Proceedings of the 1st International Workshop on Consuming Linked Data (COLD 2010), CEUR Workshop Proceedings 665, pp. 1–12. https://ceur-ws.org/Vol-665/MulwadEtAl_COLD2010.pdf

[29] Phuc Nguyen; Natthawut Kertkeidkachorn; Ryutaro Ichise; Hideaki Takeda. 2019. 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 Proceedings 2553, 1–8. https://arxiv.org/abs/1910.00246

[30] Phuc Nguyen; Ikuya Yamada; Natthawut Kertkeidkachorn; Ryutaro Ichise; Hideaki Takeda. 2020. MTab4Wikidata at SemTab 2020: Tabular Data Annotation with Wikidata. In Proceedings of SemTab 2020, CEUR Workshop Proceedings 2775, 73–78. https://ceur-ws.org/Vol-2775/paper9.pdf

[31] Phuc Nguyen; Ikuya Yamada; Natthawut Kertkeidkachorn; Ryutaro Ichise; Hideaki Takeda. 2021. SemTab 2021: Tabular Data Annotation with MTab Tool. CEUR Workshop Proceedings 3103, 1–8. https://ceur-ws.org/Vol-3103/paper8.pdf

[32] Minh Pham; Suresh Alse; Craig A. Knoblock; Pedro Szekely. 2016. Semantic labeling: a domain-independent approach. In Proceedings of ISWC 2016, LNCS 9981, Springer, 446–462. https://usc-isi-i2.github.io/papers/pham16-iswc.pdf

[33] Juan Ramos. 2003. Using TF-IDF to Determine Word Relevance in Document Queries. In Proceedings of the First Instructional Conference on Machine Learning, Rutgers University, 2003, pp. 1–7. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b3bf6373ff41a115197cb5b30e57830c16130c2c

[34] Dominique Ritze; Oliver Lehmberg; Christian Bizer. 2015. Matching HTML Tables to DBpedia. In Proceedings of WIMS 2015 (Limassol, Cyprus), 1–6 (10 pages). https://dl.acm.org/doi/10.1145/2797115.2797118

[35] Dominique Ritze; Christian Bizer. 2017. Matching Web Tables To DBpedia - A Feature Utility Study. In Proc. 20th International Conference on Extending Database Technology (EDBT), pp. 210-221. https://openproceedings.org/2017/conf/edbt/paper-148.pdf

[36] Alexey Shigarov. 2022. Table understanding: problem overview. WIREs Data Mining and Knowledge Discovery 13, 1 (2022), e1482.

https://wires.onlinelibrary.wiley.com/doi/10.1002/widm.1482?af=R

[37] Anonymous. 2024. Attribute-Based Semantic Type Detection and Data Quality Assessment. In Proceedings of IEEE/ACM BDCAT 2024, pp. 119–124. Full version will be made available upon acceptance.

[38] Anonymous. 2024. HeadersIQ – code, data, docs, and dictionaries (anonymised repository). Available at: https://github.com/HeadersIQ/HeadersIQ-main/tree/main/code

[39] Yushi Sun; Hao Xin; Lei Chen. 2023. RECA: Related Tables Enhanced Column Semantic Type Annotation Framework. Proceedings of the VLDB Endowment 16, 6 (2023), 1319–1331. https://doi.org/10.14778/3583140.3583149

[40] Bram Steenwinckel; Filip De Turck; Femke Ongenae. 2021. MAGIC: Mining an Augmented Graph using INK, starting from a CSV. In Proceedings of SemTab 2021, CEUR Workshop Proceedings 3103, 49–54 https://ceur-ws.org/Vol-3103/paper6.pdf

[41] Bram Steenwinckel; Gilles Vandewiele; Frederik De Turck; Femke Ongenae. 2019. CSV2KG:Transforming tabular data into semantic knowledge. In Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2019), CEUR Workshop Proceedings 2553, 28–35 https://ceur-ws.org/Vol-2553/paper5.pdf

[42] Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating Columns with Pre-trained Language Models. In Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22). Association for Computing Machinery, New York, NY, USA, 1493–1503. https://doi.org/10.1145/3514221.3517906

[43] Nathan Vandemoortele; Bram Steenwinckel; Sofie Van Hoecke; Femke Ongenae. 2024. Scalable Table-to-Knowledge Graph Matching from Metadata using LLMs. In Proceedings of SemTab 2024, CEUR Workshop Proceedings https://ceur-ws.org/Vol-3889/paper4.pdf

[44] Jingjing Wang; Haixun Wang; Zhongyuan Wang; Kenny Q. Zhu. 2012. Understanding tables on the web. In Conceptual Modeling (ER 2012), LNCS 7532, Springer, 141-155. https://link.springer.com/chapter/10.1007/978-3-642-34002-4_11

[45] Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çağatay Demiralp and Wang-Chiew Tan. Sato: Contextual Semantic Type Detection in Tables. PVLDB, 13(11): 1835-1848, 2020. https://www.vldb.org/pvldb/vol13/p1835-zhang.pdf

[46] Jiani Zhang; Zhengyuan Shen; Balasubramaniam Srinivasan; Shen Wang; Huzefa Rangwala; George Karypis. 2023. NameGuess: Column Name Expansion for Tabular Data. In Proceedings of EMNLP 2023, 13276–13290 https://aclanthology.org/2023.emnlp-main.820.pdf

[47] Ziqi Zhang. 2016. Effective and efficient Semantic Table Interpretation using TableMiner+. Semantic Web. 2016;8(6):921-957 https://journals.sagepub.com/doi/full/10.3233/SW-160242