A Scalable, Explainable Header-Centric Framework for Semantic Table Interpretation and Data Quality Assessment

MMMMM†
CCCCC
PPPPPP
mmmmm@mmm.mm

HHHHH
CCCCC
PPPPPP
hhhhh@hhh.hh

VVVVV
CCCCC
PPPPPP
vvvvvv@vvv.vv

ABSTRACT

Semantic Table Interpretation (STI), the process of mapping tabular data to knowledge graphs, is essential for automated data integration and quality control. Within STI, **Column Type Annotation (CTA)** assigns meaningful semantic types to table columns, supporting interoperability, analytics, and automated data quality assessment. While works like Sherlock and SATO, have downplayed header-centric approaches, recent surveys and our analysis show that many competitive systems now leverage column headers. We present a scalable, explainable, header-centric CTA framework based on the **FinalFormat** type system: a curated set of 39 semantic types for actionable data quality assessment. Our approach integrates evolving dictionaries, systematic abbreviation handling, semantic similarity, and an ML pipeline for ambiguous cases, with full traceability through **SourceKeywords**.

Our method addresses the new requirements established by **SemTab2025**, including noise robustness, alias and acronym resolution, and NIL detection for unmappable headers. We validate our framework on diverse benchmarks, mapping columns to both FinalFormat types and KG properties.

We validate our framework on a diverse suite of benchmarks, including **UCI** datasets, *Prague* relational database tables, **T2Dv2**, **VizNet** (and **Sato** subset), **Kaggle**, **SOTAB**, and the **SemTab 2024 Metadata Track**, mapping columns to both FinalFormat types and knowledge graph properties and classes from **DBpedia** and **Schema.org**. Our ***HeadersIQ*** metric provides an interpretable and actionable measure of dataset-level quality by summarizing completeness, consistency, and semantic errors across tables. Although conceived independently, this metric aligns closely with the vision of automated quality assessment outlined by **the SemTab 2024 IsGold? Track** and the data quality dimensions proposed in the **KG2Tables** 2025 study, notably diversity of semantic topics, annotation accuracy, and structural coherence.

Empirical results demonstrate high coverage and accuracy; human-in-the-loop review reveals many apparent benchmark "errors" arise from limitations in ground truth, not system failures. Our framework unites an interpretable type system, ML pipeline, enrichment, and traceability, anticipating and fulfilling the latest evaluation criteria. Robust header-centric CTA is both possible and essential for the next generation of trustworthy, automated data management.

All documents, codes, files used are in [38]

CCS CONCEPTS

- Information systems → Data management systems → Information integration → Data cleaning
- Computing methodologies → Knowledge representation and reasoning; Machine learning

KEYWORDS

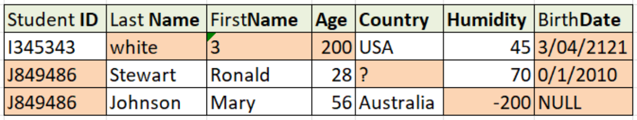
Semantic table interpretation, semantic type detection, data quality, missing data, machine learning, knowledge graphs, table understanding, column annotation

1 INTRODUCTION

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

Semantic Table Interpretation (STI), the process of mapping tables to knowledge graphs or ontologies, has emerged as a core area for enabling these downstream tasks. In particular, the **Column Type Annotation (CTA)** subtask, which assigns meaningful types to table columns, is pivotal for interoperability, data cleaning, and analytics. Throughout this work, "table" refers to any rectangular, column-row structure, whether from a database, spreadsheet, or dataset such as those from UCI or Kaggle.

Table 1.1 – Example of a dirty dataset



Student **ID**	Last **Name**	First**Name**	**Age**	**Country**	**Humidity**	Birth**Date**
I345343	white	3	200	USA	45	3/04/2121
J849486	Stewart	Ronald	28	?	70	0/1/2010
J849486	Johnson	Mary	56	Australia	-200	NULL

Motivating Example. Table 1.1 illustrates typical data quality problems - duplicates, missing values, numeric outliers, formatting inconsistencies, and type violations - that can be detected simply by analyzing the words and terms in column headers. Strikingly, many such issues can be detected by analyzing column headers alone. These header terms (in **bold**) such as "**ID**", "**Name**", "**Age**", "**Humidity**", and "**Date**" encode strong semantic expectations. Our system extracts and normalizes these signals using a large, continually evolving dictionaries (**almost 3000 terms to formats**, plus **1000 abbreviations to terms**), mapping them to 39 curated FinalFormat types, each linked to corresponding data quality rules. This enables automated, explainable, and scalable data profiling, even before value-based analysis.

Rationale and Justification for a Header-Driven Approach.

A central methodological decision in this work is the explicit focus on column headers as the primary signal for semantic type detection and data quality assessment, which departs from traditional skepticism in the field.

Historical Context and Prevailing Skepticism.

The mainstream research narrative on CTA and semantic type detection has long been shaped by skepticism toward header-centric approaches. Influential works, notably Sherlock (2019) [14] and SATO (2020) [45], both from the same research group, not only developed type detection models that deliberately excluded column headers as input but also introduced widely used benchmarks and evaluation protocols that recommended disregarding header-based features.

Their rationale was that dictionary- and header-driven strategies are fundamentally limited:

- suitable only for “simple types” (e.g., string, integer, date),
- unable to generalize to complex or context-dependent semantic types,
- reliant on static, incomplete dictionaries that are quickly outpaced by the variability and messiness of real-world data.

As a result, widely adopted benchmarks and system evaluations often advised against header-based features, either to avoid potential information leakage or due to robustness concerns, shaping prevailing community norms. Similarly, commercial tools like Tableau leverage header matching for basic type inference. However, they default to primitive types when header terms are unrecognized, sacrificing semantic richness and explainability.

Emerging Evidence: The Case for Headers

Despite this skepticism, recent years have seen a strong shift in the community. Liu et al. (2023) [23] present a comprehensive survey showing that many competitive CTA systems (at least 12 in their Table 1) now leverage header information through lookups, feature engineering, iterative matching, or deep learning. As they conclude:

“**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.”

Community Benchmarks and Data Quality

The **Semantic Web Challenges on Tabular Data to Knowledge Graph Matching (SemTab)** (since 2019) [16, 17, 8, 12, 7] is collocated with the annual **International Semantic Web Conference** and has served as the premier annual competition for benchmarking STI systems, including not only Column Type Annotation (CTA), but also Column Entity Annotation (CEA), and Column Property Annotation (CPA). The 2024 edition [12] notably introduced two key tracks: a **Metadata-Only track** allowing the focus on header-based semantic annotation to promote privacy-compliant and scalable approaches; and the **IsGold? track** [1] emphasizing automated **data quality assessment** of STI benchmark datasets by automatically identifying potential **data quality issues** including structural noise, annotation inconsistencies, and NIL misuse. Although the IsGold? track did not receive submissions, it articulated a vision for large-scale, automated quality inspection.

Abdelmageed et al. (2025) [2] in their **KG2Tables** study highlight that “**data quality remains an open gap**” in Semantic Table Interpretation research and suggest three essential **data quality dimensions**: (1) diversity of semantic topics, (2) accuracy and clarity of annotations, and (3) structural coherence.

Additionally, **the SemTab 2025 challenge** [7] now prioritizes noise robustness, alias/acronym resolution, and NIL detection for real-world data, as seen in the secu-table and MammoTab tracks. These developments reflect a broader shift toward robust, metadata-centric annotation standards.

A methodological shift is also evident in **AdaTyper** (2023) [15], also from the Sherlock/SATO research group, who previously advocated for value-only methods, but reversed course by making header analysis the first step in their type detection pipeline, using advanced semantic similarity for ontology mapping and falling back to cell-based features as needed. This validates our methodological stance and demonstrates the practical value of robust header-driven pipelines.

Motivations for a Header-Driven Pipeline.

Given these developments, the proposed methodology capitalizes on the latent semantic richness of headers, motivated by several key factors:

- **Explainability:** Headers provide compact, human-readable clues about a column’s intended meaning. The pipeline is fully traceable via SourceKeywords and dictionary mappings, supporting human-in-the-loop review.
- **Scalability and Privacy:** Annotating from headers enables efficient profiling of large datasets without inspecting sensitive or voluminous cell values, essential for privacy-conscious or open data environments.
- **Early Detection and Quality Control:** Many data quality issues (e.g., duplicate IDs, missing values, out-of-range dates, categorical inconsistencies) **can be flagged even before cell-value analysis**, simply by interpreting headers.
- **Empirical Success:** As shown in Chapter 3, the feedback-driven header-centric pipeline achieves **high coverage and accuracy across diverse datasets**, outperforming header-agnostic or value-only approaches in several domains.

Addressing the Classical Critiques

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

- **Static vs. Dynamic Dictionaries:** We employ a **large, evolving dictionary (>2700 terms),** continuously expanded through feedback loops and error logs to handle novel or noisy headers.
- **Ambiguity and Variants:** Advanced normalization, tokenization, and **abbreviation expansion (over 1000 mapped variants)** provide robust handling of misspellings, abbreviations, and synonyms.
- **Beyond Simple Types:** The **FinalFormat type system** (Section 2.4) bridges atomic types and ontological classes, achieving both coverage and interpretability, with direct links to data quality rules.

Fallback ML and NIL Assignment: Ambiguous or unmappable headers trigger machine learning pipelines and explicit NIL labeling, rather than silent misclassification.

Approach and Contributions.

This work directly address the limitations of traditional skepticism and static dictionary approaches. The dynamic, feedback-driven pipeline demonstrates that robust header-centric methods are not only competitive with, but often superior to, cell-content-based or header-agnostic systems. This establishes a new, scalable, and explainable paradigm for semantic table interpretation, supporting the next generation of automated data quality and integration tools.

This paper substantially extends a previously published attribute-based semantic type detection and data quality assessment framework [37], originally validated on 50 UCI datasets [24] with smaller dictionaries. We introduce 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**: The system is validated on diverse sources (UCI, Prague, T2Dv2, VizNet/Sato, Kaggle, SOTAB, SemTab 2024), annotated with both FinalFormat types and standard Knowledge Graph (KG) properties (DBpedia, Schema.org).
2. **Transparency and Human-Centered Evaluation**: Each type assignment is justified by SourceKeywords, enabling human-in-the-loop review. The thorough ground truth analysis reveals overlooked errors and supports the community's push for explainable annotations.
3. **Expanded Semantic Type System:** We extend the "FinalFormat" type system to cover 39 interpretable types for fine-grained, actionable annotation.
4. **Dynamic Formats and Abbreviation Dictionaries Expansion:** The feedback-driven pipeline enriches dictionaries using error logs and human feedback, significantly improving the system's robustness to variants, acronyms, and multilingual tokens.
5. **Integrated Machine Learning Pipeline:** We evaluated multiple ML models, including Random Forest and Logistic Regression, on 100 datasets (UCI and Prague), showing how fallback learning boosts performance in ambiguous or out-of-vocabulary cases.
6. **Automated Data Quality Assessment with *HeadersIQ*:** We propose a new dataset-level quality metric, *HeadersIQ*, that quantifies completeness, consistency, and semantic errors across tables, aligned with the SemTab 2024 **'IsGold ?'** vision.

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 machine learning fallback, the approach excels in both clean and noisy header scenarios, setting a new standard for scalable, resilient, and explainable column type annotation.

2 METHODOLOGY

Figure 2.1 summarizes our end-to-end pipeline for semantic type detection and annotation. The subsequent sections (2.2–2.7) describe each stage in detail, from initial ingestion and normalization through dictionary expansion, rule-based and ML assignment, to knowledge graph mapping and feedback-driven enrichment.

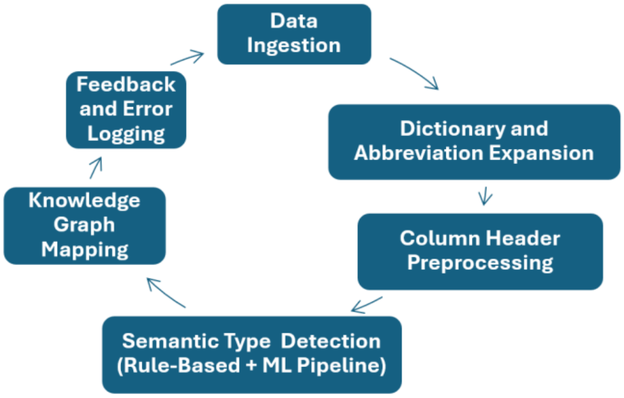

Data Ingestion
Dictionary and Abbreviation Expansion
Column Header Preprocessing
Semantic Type Detection (Rule-Based + ML Pipeline)
Knowledge Graph Mapping
Feedback and Error Logging

Figure 2.1: Pipeline Diagram for Semantic Type Detection

- **Data Ingestion:** Loading tabular data from sources such as relational databases, spreadsheets, and public datasets (e.g., UCI, Kaggle).
- **Dictionary and Abbreviation Expansion:** Employing a large, evolving dictionary of terms, abbreviations, and variants to enhance header interpretation and semantic coverage.
- **Header Preprocessing:** Normalizing, tokenizing, and expanding column headers to prepare for semantic analysis.
- **Semantic Type Detection (Rule-Based + ML Pipeline):** At the core of our approach lies the *FinalFormat Semantic*

Type System: a curated set of 39 interpretable types bridging atomic data types and complex ontologies, designed to enable both precise semantic annotation and actionable data quality rules.

The type assignment is performed via a hybrid pipeline combining deterministic rule-based matching with a fallback machine learning module. Rules capture exact and fuzzy dictionary matches on normalized headers, while the ML pipeline resolves ambiguous or previously unseen cases using features extracted from headers. This combination ensures robust, scalable, and explainable semantic typing.

- **Knowledge Graph Mapping:** Linking FinalFormat assignments to standard properties and classes from DBpedia and Schema.org to facilitate semantic interoperability.
- **Feedback and Error Logging:** Capturing assignment errors, unknown terms, and ambiguous mappings to enable iterative dictionary and model refinement through human-in-the-loop review.

This design supports extensibility, transparency, and adaptability to new domains and noisy, real-world data.

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. Importantly, the ingestion process supports iterative reprocessing triggered by feedback from subsequent stages, enabling continual refinement of semantic annotations.

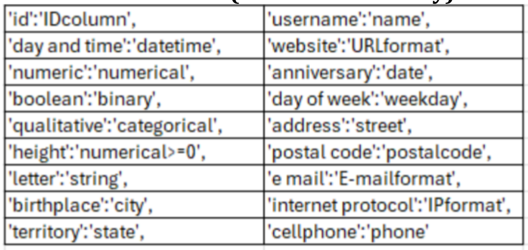
2.2 Dictionary and Abbreviation Expansion

A key component of our research is a comprehensive and continuously evolving **formats dictionary** that maps a diverse set of header tokens, including synonyms, variants, and domain-specific terms, to a curated set of 39 FinalFormat semantic types (to be detailed in Section 2.4). This dictionary is foundational for robust, explainable semantic type detection, enabling our system to handle noisy, abbreviated, or unconventional headers with high resilience.

Format Dictionary Creation and Expansion

In previous work [37], the formats dictionary was built using words and terms from the Web Data Commons project [21] which analyzed over 90 million tables in the Web Table Corpus [5] and linked column headers to the DBpedia entities (DBHeaders file [9])This process produced an initial dictionary of over 1,100 unique header terms. The dictionary has since been expanded to 1800 using ChatGPT-based term suggestions[37], and later, with manual curation, and systematic feedback after analysis of over 120,000 columns from many data sources to the new total of almost 2800 unique header tokens mapped to 39 FinalFormat types, with some illustrated in Table 2.1.

Table 2.1 - Example mappings from header terms to FinalFormats (formats dictionary)



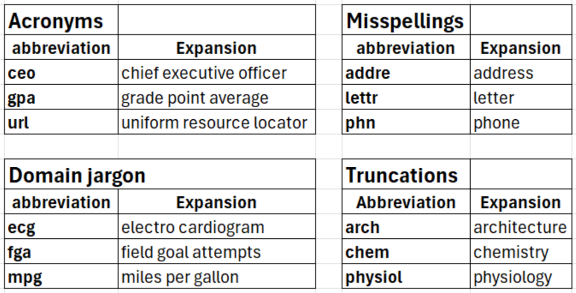
'id':'IDcolumn',	'username':'name',
'day and time':'datetime',	'website':'URLformat',
'numeric':'numerical',	'anniversary':'date',
'boolean':'binary',	'day of week':'weekday',
'qualitative':'categorical',	'address':'street',
'height':'numerical>=0',	'postal code':'postalcode',
'letter':'string',	'e mail':'E-mailformat',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

Table 2.1 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 1000 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" → "percentage"). Table 2.2 presents representative examples from the abbreviations dictionary, illustrating the breadth and diversity of normalized forms and their mappings, including coverage of **acronyms, domain jargon, misspellings and truncations** (such as "abdnt" → "abandonment", "advsry" → "advisory" derived from NameGuess[46], which expands abbreviations in column names to their full forms (e.g., "dob" → "date of birth") using a neural model trained on tables.

Table 2.2 - Acronyms, domain jargon, misspellings and truncations



Acronyms		Misspellings	
abbreviation	Expansion	abbreviation	Expansion
ceo	chief executive officer	addre	address
gpa	grade point average	lettr	letter
url	uniform resource locator	phn	phone
Domain jargon		**Truncations**	
abbreviation	Expansion	Abbreviation	Expansion
ecg	electro cardiogram	arch	architecture
fga	field goal attempts	chem	chemistry
mpg	miles per gallon	physiol	physiology

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. This strategy equips the system to reliably handle corrupted, partial, or novel headers, as required for the secu-table and MammoTab tracks.

Multilingual and Cross-Domain Support

While most tokens are English, we are actively extending multilingual coverage (e.g., Portuguese, Spanish), using language-specific variant inclusion, to address global tabular diversity.

Data Sources and Domain Breadth

Dictionaries are built after analysis of authoritative sources, including UCI datasets [24], Prague tables [26], T2Dv2 [35], VizNet [13], Sato [45], Kaggle [18], and SOTAB [20], ensuring practical coverage across domains and real-world settings.

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, forming the semantic evidence for type assignment.

2.4 FinalFormat Semantic Type Detection System

This corresponds to the Semantic Type Detection stage from Figure 2.1. The FinalFormat semantic type system is the backbone of our approach, providing a compact yet expressive set of 39 interpretable types that bridge atomic data types (such as string, integer, date) and complex, fragmented knowledge graph ontologies like DBpedia [4] and Schema.org [20]. FinalFormat is designed to deliver actionable, explainable, and scalable semantic type assignments, specifically tailored for automated data quality assessment in tabular data.

Motivation

Atomic types are too generic for nuanced quality checks, while direct mappings to large ontologies are often inconsistent, overly complex, and hard to maintain. FinalFormat provides a **middle ground**: a curated set of 39 semantic types that are expressive enough for practical data quality profiling, yet compact and interpretable for transparency and operational consistency.

Design Principles

- **Coverage**: Captures the most frequent and important semantic patterns in real-world tables, across many domains and sources.
- **Manageability**: Maintains a practical set of types, avoiding fragmentation and minimizing annotation ambiguity.
- **Actionability**: Each type is directly linked to domain-specific quality rules (e.g., uniqueness, range, format validation), enabling instant automated profiling.
- **Extensibility**: Modular, feedback-driven updates allow rapid incorporation of new types based on error analysis and domain evolution.

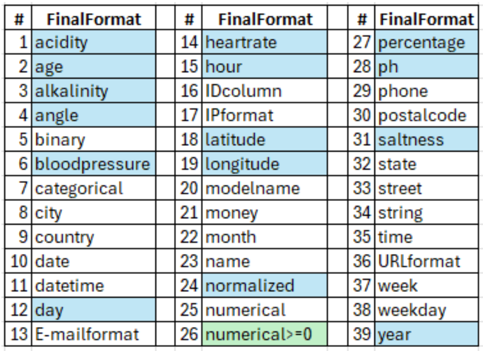
Definition and Scope

FinalFormat types cover categories such as:

- **Numerical**: generic numeric, non-negative, and bounded numericals (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”, “is”, “or not”).
- **Financial**: money
- **Textual**: descriptive, free text, assigned to string.

Table 2.3 shows the 39 FinalFormats, highlighting the non-negative numbers (which is a unique way to evaluate data) and some **numerically bounded formats (in blue)**, such as **'percentage', age',** and **domain-specific types** such as **'heartrate'** ,**'latitude'** and '**ph**'.

Table 2.3 – The 39 FinalFormats



#	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

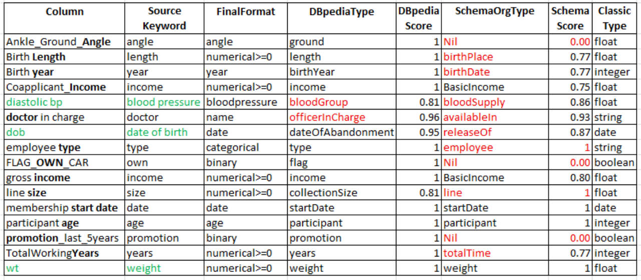
Mapping to Knowledge Graphs and Comparison to Atomic Types and Ontologies

This corresponds to the Knowledge Graph Mapping stage from Figure 2.1.

Once a FinalFormat type is assigned to a column, our system seeks the most semantically aligned property in DBpedia or Schema.org. This is accomplished using both string similarity and embedding-based measures, enabling precise links between interpretable types and knowledge graph classes/properties. The resulting mappings enhance interoperability and facilitate further semantic integration of tabular data

Table 2.4 illustrates a real example of our end-to-end FinalFormat semantic type assignment process, detailing header normalization, abbreviation expansion, dictionary matching, fallback machine learning predictions, and SourceKeywords extraction. Notably, ontology mappings to DBpedia [4] and Schema.org [20] properties are integrated seamlessly within this pipeline, with confidence scores guiding assignment reliability. Also present are classic atomic data types. This illustrates the system's finer semantic granularity and greater practical value for data quality profiling.

Table 2.4 – Semantic Type Assignment for FinalFormat, DBpedia and Schema.org



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

Observe that the Knowledge Graph Ontologies annotations (DBpedia [4] and Schema.org [20]) took advantage of the expansion of the abbreviated terms, such as bp, dob and wt, but even then, they didn't bring appropriate values sometimes (see bloodGroup for DBpedia, and many items for Schema.org).

Matching and Annotation Algorithms

Our semantic type assignment process combines deterministic rule-based matching with a robust machine learning (ML) (Figure 2.1) pipeline to maximize accuracy, coverage, and explainability.

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. Additionally, semantic similarity is employed via FastText embeddings and cosine similarity to compare headers with ontology labels, enhancing the matching process by capturing lexical nuances.

Machine Learning Pipeline

For columns where rule-based matching yields ambiguous or unresolved results, an ML pipeline acts as a fallback to improve coverage and resolve uncertainty. The detailed algorithm is presented in Algorithm 1. Key components include:

- **Preprocessing and Feature Engineering**

Column headers and descriptions undergo preprocessing steps detailed in Section 2.3, including normalization, abbreviation replacement, camel case splitting, and tokenization. The resulting textual data, cleaned headers, descriptions, and extracted SourceKeywords, is vectorized with a TF-IDF model [33] using the combined vocabulary from DBHeaders [9] and the Formats dictionary. This representation emphasizes domain-specific terms, enhancing classification accuracy.

- **Addressing Class Imbalance**

Semantic type classification suffers from class imbalance due to varying frequencies of types. To mitigate this, we apply Synthetic Minority Over-sampling Technique (SMOTE) to generate synthetic minority class samples, ensuring better model generalization without overfitting

- **Model Selection and Hyperparameter Optimization** We train multiple classifiers (Random Forest, Logistic Regression, Gradient Boosting, K-Nearest Neighbors, LinearSVC) and optimize hyperparameters using GridSearchCV with stratified cross-validation. This ensures fair tuning across imbalanced classes.
- **Evaluation and Interpretation**

Models are evaluated using precision, recall, and F1-score metrics on held-out test data. Feature importance analysis further aids interpretability.

- **Model Persistence**

The best-performing models and the TF-IDF vectorizer [33] are saved to support reproducibility and deployment in production environments.

Headers that cannot be confidently mapped are explicitly assigned a NIL type and logged for manual review and dictionary enrichment, which reduces unknown rates and increases resilience to ambiguous or missing metadata.

Confidence Scoring and Ambiguity Management

Each assignment is associated with a confidence score derived from either rule matching strength or ML prediction probabilities. This enables effective ranking of candidate types and supports ambiguity resolution strategies including fallback logic and human-in-the-loop interventions.

Feedback and Iterative Enrichment

Our system incorporates error logging to track unknowns and mismatches, guiding dictionary expansion and rule refinement. Human-in-the-loop review is employed for correcting and enriching dictionaries, especially for novel or ambiguous cases. This continuous improvement cycle is critical for adapting the system to new datasets and real-world data variability.

Inspired by the SemTab 2024 **IsGold? Track** [1] and the KG2Tables 2025 study [2], our feedback pipeline prioritizes detection and correction of structural noise, annotation inconsistencies, NIL misuse, and semantic incoherence. This approach ensures systematic, large-scale enhancement of semantic type detection and automated data quality inspection.

2.5 Integration with Data Quality Assessment

As detailed in our previous work [37], our framework links semantic type assignments directly to automated data quality rules, enabling early and explainable profiling of tabular data. Each FinalFormat semantic type triggers a tailored set of quality checks, such as uniqueness constraints, valid ranges, and format validations, allowing immediate detection of data anomalies aligned with the column's inferred meaning.

To summarize dataset-level quality, we introduce HeadersIQ (defined in Section 3.8), an interpretable metric that aggregates detected data quality issues across all columns for each table. HeadersIQ quantifies three core dimensions of data quality: completeness, semantic consistency, and structural coherence, reflecting the latest priorities highlighted by recent benchmark challenges such as SemTab and KG2Tables [2].

This approach supports scalable and automated data quality inspection over large and diverse data sources, including those with noisy or corrupted metadata. By leveraging rich semantic typing coupled with domain-specific quality rules, HeadersIQ provides actionable insights that facilitate continuous monitoring and prioritization of data cleaning efforts.

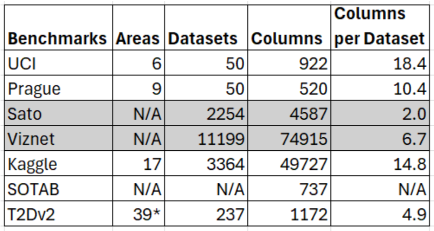
3 RESULTS AND COVERAGE ANALYSIS

This chapter presents a comprehensive evaluation of our 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.

3.1 Datasets and Benchmarking Scope

To rigorously assess the scalability, robustness, and generalizability of our header-centric semantic type annotation pipeline, we evaluated it across a broad set of well-established benchmarks and real-world datasets. These include classic datasets and relational database tables (UCI [24], Prague [26]), large-scale web table corpora (Sato [45], VizNet [13]), user-contributed data repositories (Kaggle [18]), knowledge-graph-oriented benchmarks (T2Dv2 [35], SOTAB [20]), and more. Table 3.1 summarizes the structural diversity, coverage, and scale of each benchmark.

Table 3.1 – Structural Statistics of Benchmarks



Benchmarks	Areas	Datasets	Columns	Columns per Dataset
UCI	6	50	922	18.4
Prague	9	50	520	10.4
Sato	N/A	2254	4587	2.0
Viznet	N/A	11199	74915	6.7
Kaggle	17	3364	49727	14.8
SOTAB	N/A	N/A	737	N/A
T2Dv2	39*	237	1172	4.9

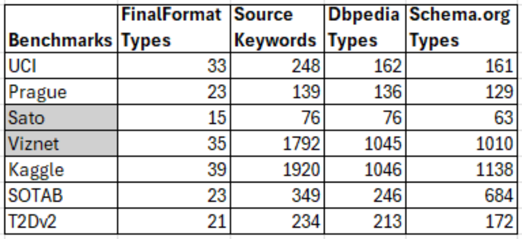
These are the Data Sources evaluated in this work:

- UCI [24] (50 classic relational datasets, high column-to-dataset ratio, real-world domains)
- Prague Relational Learning Repository [26] (50 tables, high column-to-dataset ratio, real-world domains)
- Sato [45] and VizNet [13] (large-scale web table corpora with thousands of datasets; Sato is a subset of VizNet)
- Kaggle [18] (broad, real-world user-generated datasets, highly diverse in topic and structure, 17 distinct areas)
- T2Dv2 [35] (Wikipedia tables, with DBpedia as KG reference, the 39 "Areas" in the table are indeed DBpedia Classes)
- SOTAB [20] (The WDC Schema.org Table Annotation Benchmark) Schema.org-oriented benchmark, with two annotation tasks: Properties and Expected Types)

3.2 Semantic Type Detection: Coverage and Diversity

We evaluated the breadth and diversity of semantic types discovered by our system in each benchmark, comparing our compact FinalFormat set, SourceKeyword diversity, and the type coverage of two reference KGs (DBpedia [4], Schema.org [20]). Table 3.2 and Figure 3.1 present the results.

Table 3.2: Coverage of FinalFormat, SourceKeywords, and KG Types



	FinalFormat	Source	Dbpedia	Schema.org
Benchmarks	Types	Keywords	Types	Types
UCI	33	248	162	161
Prague	23	139	136	129
Sato	15	76	76	63
Viznet	35	1792	1045	1010
Kaggle	39	1920	1046	1138
SOTAB	23	349	246	684
T2Dv2	21	234	213	172

The FinalFormat system covers all 76 Sato formats using just 15 types (100% match), demonstrating compactness and generalization. These Sato formats come from the Sherlock paper, and in their papers they count 78 formats, but two are missing in the data ('File size' and 'Team name'). Table 3.4 will define the relation between the two systems.

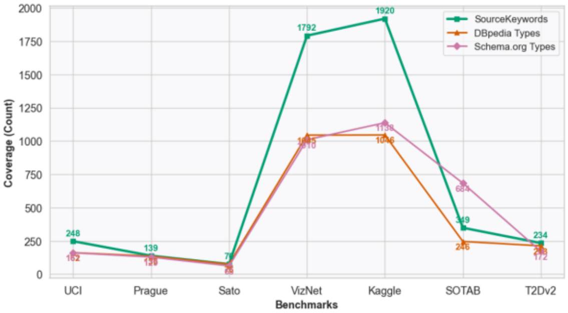
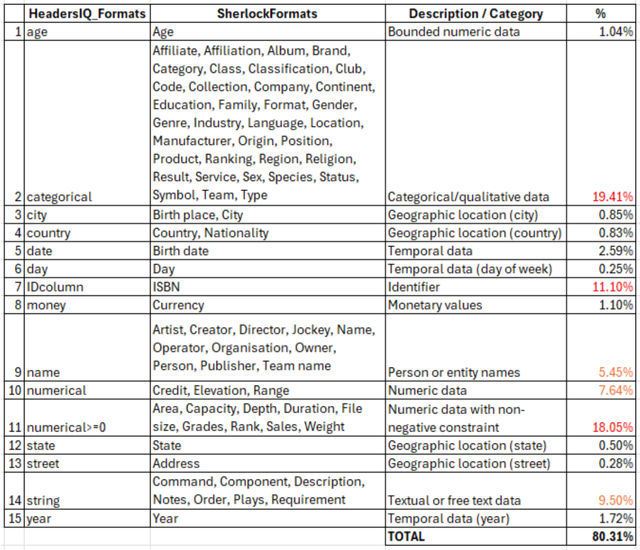


Figure 3.1: Coverage Comparison: SourceKeywords, DBpedia, Schema.org

SourceKeywords coverage consistently matches or exceeds those of DBpedia and Schema.org in all benchmarks, even as the number of KG types varies by benchmark and ontology.

Remarkably, our condensed set of 15 FinalFormats covers approximately 80% of the 118,639 unique columns analyzed from the grouping of Kaggle and Viznet datasets (Table 3.4). This coverage indicates that a smaller, well-curated set of semantic types can effectively represent the diverse real-world data types encountered in practice, facilitating scalable and interpretable type annotation. This analysis reveals a highly skewed distribution of semantic types. The majority fall into a handful of core categories, including categorical (19.41%), non-negative numerical (numerical>=0 with 18%), IDcolumn (11%), string (9.5%), and plain numerical (7.64%). This concentration confirms that effective handling of these types is critical for practical applications.

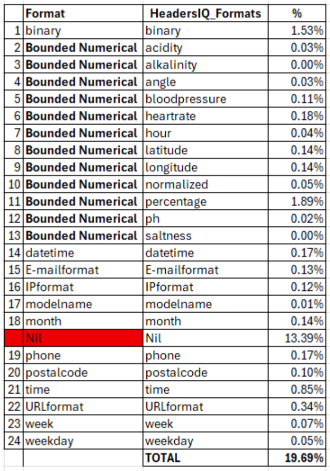
Table 3.4 – 15 FinalFormats that map all 76/78 Sherlock/Sato Formats



	HeadersIQ_Formats	SherlockFormats	Description / Category	%
1	age	Age	Bounded numeric data	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	Categorical/qualitative data	19.41%
3	city	Birth place, City	Geographic location (city)	0.85%
4	country	Country, Nationality	Geographic location (country)	0.83%
5	date	Birth date	Temporal data	2.59%
6	day	Day	Temporal data (day of week)	0.25%
7	IDcolumn	ISBN	Identifier	11.10%
8	money	Currency	Monetary values	1.10%
9	name	Artist, Creator, Director, Jockey, Name, Operator, Organisation, Owner, Person, Publisher, Team name	Person or entity names	5.45%
10	numerical	Credit, Elevation, Range	Numeric data	7.64%
11	numerical>=0	Area, Capacity, Depth, Duration, File size, Grades, Rank, Sales, Weight	Numeric data with non-negative constraint	18.05%
12	state	State	Geographic location (state)	0.50%
13	street	Address	Geographic location (street)	0.28%
14	string	Command, Component, Description, Notes, Order, Plays, Requirement	Textual or free text data	9.50%
15	year	Year	Temporal data (year)	1.72%
			TOTAL	**80.31%**

Additionally, our system includes 24 supplementary semantic types, not handled by Sherlock's Formats, predominantly bounded numerical formats, such as health related (heartrate (0.18%), and bloodpressure (0.11%)), percentage(1.89%) and chemical related such as acidity(0.03%) and ph (0.02%), or geographically related (latitude and longitude (0.14%)), enabling more nuanced detection and assessment of data quality issues that traditional taxonomies omit. These add up to around 6%, totalling over 7000 columns in this universe, which is quite considerable.

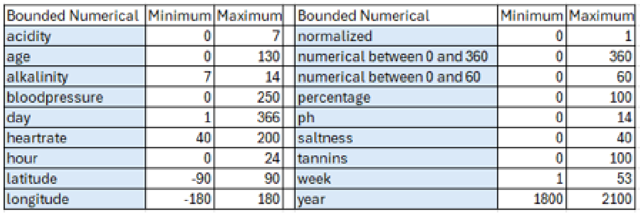
Table 3. 5 Additional Formats that are not considered by Sherlock/Sato



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

The Nil category represents a notable challenge, encompassing columns without confident type assignment, often due to missing or noisy headers. Our iterative dictionary expansion and machine learning fallback pipeline successfully reduce these cases, enabling improved coverage and higher data quality awareness. But it is totalling only 13.39% of Nil values (86.61% of Valid cases in the grouping of the two largest data sources). A real low amount considering over 118000 columns analysed. The table 3.6 shows the minimum and maximum values adopted of the Bounded Numerical types.

Table 3. 6 Bounded Numerical types

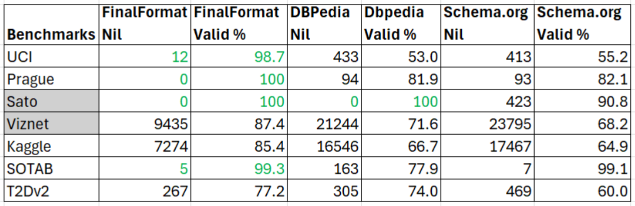


Bounded Numerical	Minimum	Maximum		Bounded Numerical	Minimum	Maximum
acidity	0	7		normalized	0	1
age	0	130		numerical between 0 and 360	0	360
alkalinity	7	14		numerical between 0 and 60	0	60
bloodpressure	0	250		percentage	0	100
day	1	366		ph	0	14
heartrate	40	200		saltness	0	40
hour	0	24		tannins	0	100
latitude	-90	90		week	1	53
longitude	-180	180		year	1800	2100

3.3 Validity, NIL Rates, and Robustness

Table 3.7 compares the rates of valid type assignments and the prevalence of NIL (unmappable) cases across systems. This direct, quantitative comparison demonstrates the effectiveness of our approach in maximizing type coverage, minimizing unmapped columns, and outperforming established KG-based methods.

Table 3.7 NIL/Valid Assignment Rates



Benchmarks	FinalFormat Nil	FinalFormat Valid %	DBPedia Nil	Dbpedia Valid %	Schema.org Nil	Schema.org Valid %
UCI	12	98.7	433	53.0	413	55.2
Prague	0	100	94	81.9	93	82.1
Sato	0	100	0	100	423	90.8
Viznet	9435	87.4	21244	71.6	23795	68.2
Kaggle	7274	85.4	16546	66.7	17467	64.9
SOTAB	5	99.3	163	77.9	7	99.1
T2Dv2	267	77.2	305	74.0	469	60.0

Key Observations:

- FinalFormat assignment is always at least as good as, and usually far better than, KG-based approaches (DBpedia, Schema.org) in terms of "Valid %" and low NIL rates.

- Even on large and noisy corpora (**Kaggle, VizNet**), our system achieves robust results (85–87% FinalFormat valid assignments, see also Figure 3.2).
- **Sato** and **Prague** demonstrate 100% valid assignment, showing the completeness of our dictionary and rules for structured, curated benchmarks.
- Our coverage is particularly impressive on **UCI**, which has much higher values than the ones on DBpedia and Schema.org.

3.4 Machine Learning Semantic Type Detection System

We evaluated five Machine Learning models (Random Forest, Gradient Boosting, K-Nearest Neighbors, Logistic Regression, and Linear-SVC), following the methodology described in Chapter 2, which involves preprocessing (TF-IDF vectorization [33], SMOTE for imbalance handling) and hyperparameter tuning (GridSearchCV with stratified cross-validation).

All models were trained on an 80–20 train-test split of the data. We specifically focused on handling class imbalance, particularly for minority semantic types such as 'URL format' or 'time', using SMOTE applied exclusively on training data to prevent data leakage.

The detailed results for precision, recall, and F1-score across different semantic formats are illustrated in Figures 3.3, 3.4, and 3.5.

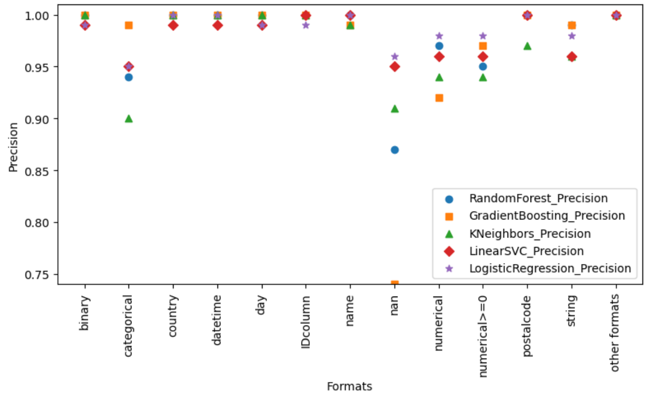


Figure 3.3: Precision for Formats on 5 ML Models

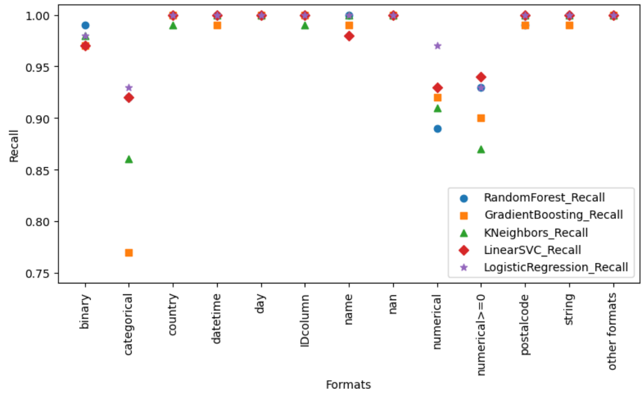


Figure 3.4: Recall for Formats on 5 ML Models

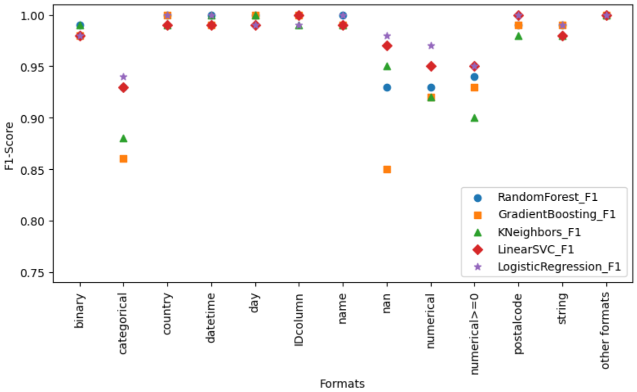


Figure 3.5: F1-Score for Formats on 5 ML Models

- **Categorical Format**: Showed variability across models. GradientBoosting in particular struggled, with a recall of 0.77 and F1-score of 0.86. This suggests that categorical types are inherently harder to distinguish from others, especially in noisy or complex tables.
- **Numerical vs Numerical>=0**: These types often presented confusion. F1-scores ranged from 0.89 to 0.97 for 'numerical' and from 0.90 to 0.98 for 'numerical>=0'. This indicates that separating general numerical values from non-negative constraints remains a challenge for some classifiers.
- **Nan/Nil Format**: Represents missing or undefined format. While LogisticRegression (F1: 0.98) and LinearSVC (F1: 0.97) handled these cases well, GradientBoosting struggled with a precision of only 0.74, misclassifying ambiguous headers more frequently.
- **Formats with Perfect Scores Across All Models**: Formats such as E-mailformat, URLformat, Age, City, Country, Date, Hour, Latitude, Longitude, Modelname, Month, Percentage, Ph, Phone, State, Street, Time, Weekday, Year, Normalized, and Postalcode achieved perfect precision, recall, and F1-scores across all five models, demonstrating reliability in identifying structured, well-labeled attributes.

Model-Specific Insights:

The average precision, recall, F1-score, and accuracy for the five Machine Learning models are summarized in Figure 3.6:

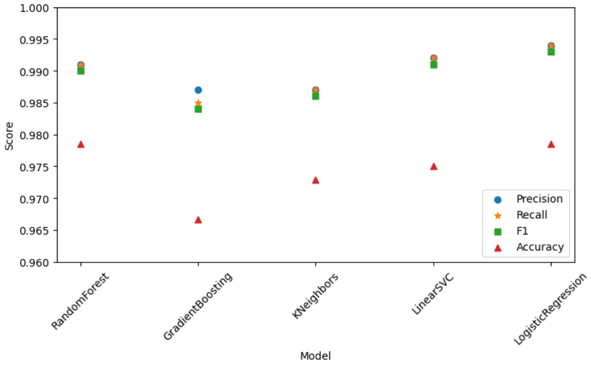


Figure 3.6: Average Metrics on 5 ML Models

3.5 Handling Complex and Ambiguous Header Cases

Our 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 3. 8 Detailed FinalFormat and SourceKeywords allocation

Original Column	Cleaned Column	SourceKeywords	FinalFormat
Members with age 5 - 17 years old	members with **age** 5 17 years old	age	age
Ankle_Ground_**Angle**	ankle ground **angle**	angle	angle
has_birth_date	**has** birth date	**has**	**binary**
IsBasedOnRealStory	**is** based on real story	**is**	**binary**
BookedHotel**OrNot**	booked hotel **or not**	**or not**	**binary**
bp	**bp**	**blood pressure**	bloodpressure
trestbps	**trestbps**	**blood pressure**	bloodpressure
Pclass_1	p**class** 1	class	categorical
ATTEND_**DEPT**	attend **dept**	**department**	categorical
Date **Range***	**date range***	range	categorical
price_**range**	**price range**	range	categorical
Father's **Birth Place**	father's **birth place**	**birth place**	**city**
Capital**Name**	**capital** name	**capital**	**city**
Author, **Country**	author, **country**	country	country
team_one_player_one_**nationality**	team one player one **nationality**	nationality	country
BIRTHDATE	birth**date**	birthdate	date
dtRef	**dt** ref	**date**	date
Data Last **Updated**	data last updated	**updated**	**date**
Request **timestamp**	request timestamp	**timestamp**	**datetime**

Table 3.8 provides a list of examples for the allocation of FinalFormats and SourceKeywords from a superset of Kaggle [18] and Viznet [13].

For example, binary-type columns are often signaled by common words such as “has”, “is”, or phrases like “or not”, which trigger classification into the binary FinalFormat. This allows for accurate detection of boolean attributes without 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 “nationality” are consistently mapped to the country FinalFormat. Similarly, “updated” fields are recognized as date types, and “timestamp” keywords indicate datetime types.

A novel addition is the recognition of medical terms such as “trestbps”, 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 attribute semantics. This is especially critical for heterogeneous, real-world datasets where inconsistent or abbreviated headers are commonplace. Appendix 8.1 describes this process in more detail.

3.6 Comprehensive Benchmarking: SemTab 2024

Our systematic analysis of semantic annotation for the SemTab 2024 Metadata-only challenge [12] initially identified semantic types correctly for only 64 out of 141 evaluated columns (0.45 accuracy, Top Hit@1), and 4 more columns at Top Hit@5 (68/141 = 0.48), reflecting the strict adherence to provided ground truth (GT).

However, after detailed human-in-the-loop review (described in Appendix 8.2), we identified an additional 43 columns as correctly annotated increasing overall accuracy to 0.76 (Top Hit@1, correct only) and 0.79 (Top Hit@5, correct only)

Considering 6 more columns as contextually appropriate annotations, we increased accuracy to 0.80 (Top Hit@1including appropriate) and 0.83 (Top Hit@5including appropriate).

According to Table 3 from Results of SemTab 2024 [12], we managed to outperform at Top Hit@1 all the only two competitors, Adwan [43] (Vandemoortele et al. 2024) and Metalinker/CVA [25]. And at Hit@5, we outperformed Metalinker/CVA but lost to Adwan. Table 3.9 shows that.

Table 3.9 Comparative Analysis for the SemTab 2024 Metadata track

Competitor	Top Hit@1	Top Hit@5
Adwan[43]	0.75	**0.92**
MetaLinker/CVA[25]	0.55	0.70
Our Method (GT Strict)	0.45	0.48
Our Method (Correct only)	**0.76**	0.79
Our Method (Correct+Appropriate)	**0.80**	0.83

3.7 *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 attribute labels (column headers or names) and its focus on Information Quality (IQ), while also evoking the idea of intelligent assessment, inspired by the concept of Intelligence Quotient.

Definition: Let

E = total number of identified Data Quality Issues (DQIs)

T = total number of data items (rows × columns)

Then:

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

Interpretation:

A score of 100 indicates a fully clean dataset (E = 0). Scores below 100 reflect the proportion of values that violated the expected rules or formats.

It provides a single-value summary that is both interpretable and comparable across datasets, enabling ongoing monitoring, prioritization of cleaning efforts, and continuous data quality improvement.

Example:

For dataset 45 with:

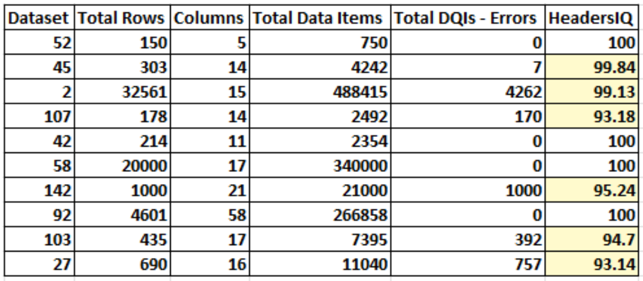
303 rows, 14 columns → 4242 data items

7 total Data Quality Issues (Errors)

HeadersIQ ≈ 99.84

This indicates that 99.84% of all values in the dataset are clean, based on the identified issues. Table 3.9 illustrates 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 3.10 - *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	4242	7	99.84
2	32561	15	488415	4262	99.13
107	178	14	2492	170	93.18
42	214	11	2354	0	100
58	20000	17	340000	0	100
142	1000	21	21000	1000	95.24
92	4601	58	266858	0	100
103	435	17	7395	392	94.7
27	690	16	11040	757	93.14

HeadersIQ reflects the proportion of clean data items in each dataset, computed as , where E is the total number of detected data issues and T is the total number of data items.

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

4 DISCUSSION

This chapter discusses the broader implications of our findings, revisits core design decisions, and critically assesses the contributions and limitations of our approach. It synthesizes insights across all previous chapters to articulate the impact and future potential of our scalable, explainable, header-centric framework for semantic table interpretation and data quality assessment.

4.1 Reevaluating the Value of Header-Centric Type Detection

Historically, header-based semantic annotation was often dismissed as unreliable, with seminal works like Sherlock [14] and SATO [45] promoting value-driven methods. However, our findings decisively counter this narrative. Through systematic preprocessing, an expansive and dynamically updated formats dictionary, and fallback ML mechanisms, we demonstrate that column headers are rich semantic indicators. This repositions headers as essential metadata elements, capable of driving precise semantic annotation and scalable data profiling.

Our FinalFormat system, grounded in 39 interpretable semantic types, bridges the gap between overly broad atomic types and highly granular ontology classes. This intermediate granularity proved effective for both human interpretability and automated quality rule execution, enabling a new paradigm in data quality assessment.

4.2 Strength, Scale, and Diversity in Format Assignment

One of the clearest demonstrations of scalability in our approach stems from the evolution of our dictionary infrastructure. During early experiments [37] on 50 UCI datasets (approximately 1,000 columns), our formats dictionary included around 1,800 semantic terms and 300 abbreviations, totaling 2,100 entries. This yielded near-perfect semantic type coverage at small scale.

To support scalability, we expanded the formats dictionary to 2,800 semantic terms and the abbreviation dictionary to 1,000 items, a combined growth of just 80%. Despite this modest expansion, the system was able to generalize over 120,000 columns collected from large-scale, heterogeneous repositories. We identified semantic types for 85% of these columns, over 102,000, demonstrating a remarkable 102× increase in practical coverage capacity. This contrast between a small dictionary growth and exponential output gain is direct evidence of the reusability and scalability of our approach, besides the huge robustness of our pipeline.

Our semantic type detection system was deployed over two real-world big data repositories, VizNet [13] and Kaggle [18], which represent diverse domains (17 different domains in almost 50,000 columns in Kaggle and 75,000 columns in VizNet), noisy headers, and unstructured metadata. Despite this heterogeneity, only 15% of columns were assigned NIL, affirming the robustness of our pipeline.

In terms of semantic expressiveness, the FinalFormat taxonomy proved effective in generalizing and extending the semantic coverage of benchmarks like Sato [45]. While covering all 76 Sato types, in only 15 of our FinalFormats, our set of 39 FinalFormats also includes extra 24 new types across key domains such as chemistry, geolocation, health, and environmental data summing up 6% of all 120,000 columns analysed in the grouped Kaggle+Viznet superset. This shows that a moderate, human-readable type system can capture broad domain diversity without requiring an explosion in type granularity.

4.3 Benchmark Comparisons and Ontology Mapping

We evaluated our semantic assignments against established knowledge graphs (DBpedia and Schema.org) and across widely recognized benchmarks, including UCI [24] (50 datasets, 900 columns), Prague [26] (50 database tables, 500 columns), Sato [45] (2250 unique tables and 4600 columns), SOTAB [20] (700 columns), and T2Dv2 [35] (1100 columns). 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.

4.4 Machine Learning Performance and Format-Specific Insights

The ML-based fallback mechanism showed particularly strong performance in distinguishing well-defined types and mitigating ambiguity. Our results indicate:

- Categorical types exhibit model variability, with Gradient Boosting struggling (Recall: 0.77, F1: 0.86), reflecting inherent challenges in delineating categorical features in noisy contexts.
- Differentiating between 'Numerical' and 'Numerical>=0' remains challenging, with F1-scores ranging from 0.89–0.98, indicating the subtlety of distinguishing semantic constraints like positivity.
- 'Nan' (undefined format) detection varied by model, with Logistic Regression (F1: 0.98) and Linear SVC (F1: 0.97) outperforming Gradient Boosting (Precision: 0.74).

Formats such as E-mailformat, URLformat, Age, City, and Country achieved perfect scores across all five models, affirming the stability of our approach on well-structured headers.

4.5 HeadersIQ: A Simple Yet Powerful Metric

The *HeadersIQ* metric emerged as a practical tool for summarizing dataset-level quality. Its interpretable formula - quantifying the proportion of valid data items - enabled rapid comparison across sources. Unlike traditional profiling tools, *HeadersIQ* reflects semantic integrity, rule violations, and consistency, offering an intelligent and domain-aware summary of data quality.

With values ranging from 93.14 to 100 in the first ten datasets evaluated, *HeadersIQ* has proven both stable and sensitive, suitable for automated monitoring in large-scale pipelines. It reflects issues that simpler tools overlook, such as the misuse of binary fields, non-normalized categorical values, or ambiguous dates.

4.6 Redefining Ground Truth and Benchmarking Practices

Our SemTab 2024 Metadata analysis revealed discrepancies between system predictions and the official ground truth. Human-in-the-loop review showed that many “errors” were in fact valid alternate annotations, raising critical questions about how ground truth is defined and evaluated.

For instance, columns initially marked incorrect by the benchmark were reassessed and deemed contextually appropriate or even superior upon manual inspection. This suggests that future benchmarks must accommodate multiple plausible answers per column and prioritize contextual correctness over rigid mappings.

4.7 Limitations and Remaining Challenges

Despite significant advances, several limitations remain:

- **Metadata Dependency**: Our system relies on meaningful headers. In cases of cryptic, sparse, or missing labels, the accuracy of semantic assignments—and downstream DQ assessment—declines.
- **Multilingual Coverage**: While partial support exists for non-English terms, comprehensive multilingual capabilities (e.g., for compound terms, domain-specific jargon) are still limited.
- **Long-Tail Semantics**: Rare domain-specific types (e.g., alkalinity, modelname) remain hard to detect due to data sparsity.
- **Unit Handling**: Our current pipeline identifies numerical formats but lacks integrated logic for detecting or validating units (e.g., kg, cm^2).

These are active areas for expansion, particularly through integration with value-based and context-aware analysis.

4.8 Future Directions

Several promising avenues emerge:

- **Multilingual Enrichment**: Extend dictionaries with translations and regional variants to improve coverage for global datasets.
- **Context-Aware Detection**: Combine header-based and value-based signals to improve disambiguation, especially for sparse or overloaded headers.

- **LLM Integration**: Use transformer-based models (e.g., BERT, GPT) to interpret ambiguous headers, boost synonym handling, and detect latent patterns.

- **Unit Normalization**: Develop rules or learning-based modules to detect, validate, and normalize measurement units.

4.9 Broader Impacts and Practical Adoption

Our framework has immediate applications across data integration, metadata management, ETL automation, and knowledge graph construction. Its explainability, scalability, and adaptability align well with industrial needs for trustworthy data systems. The integration of semantic understanding and quality assessment into a single pipeline sets a precedent for future STI systems.

By uniting dynamic dictionaries, robust header analysis, explainable type assignments, and actionable quality metrics, this work lays the foundation for intelligent metadata-driven governance in large-scale, heterogeneous data environments.

5 RELATED WORK

This section reviews prior research in Semantic Table Interpretation (STI) and Column Type Annotation (CTA), contextualizing our 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 elements - cells, columns, and rows - to entities, types, and properties in a knowledge graph (KG). Key STI subtasks include:

- **Column Type Annotation (CTA):** Assigning each column to one or more KG types.
- **Cell Entity Annotation (CEA**): Linking cell values to KG entities.
- **Column Property Annotation (CPA):** Identifying relationships between columns.
- **Row-to-Instance Linking (R2I)** and **Topic Annotation (TA)** are also common extensions.

Recent **SemTab** challenges have introduced metadata-only tracks, such as SemTab 2024's Metadata-to-KG task, which require systems to infer semantic types without access to data values. This context motivates scalable and interpretable methods based solely on metadata, including column headers.

5.2 Column Type Annotation Paradigms

Based on Liu et al.'s taxonomy (From their Table 1), we identify three broad families of CTA systems:

- **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**: 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].

Liu et al.'s comparative framework evaluated over 30 systems across dimensions such as supported tasks (CEA, CTA, CPA), metadata utilization, KG alignment (e.g., DBpedia, Wikidata), and benchmark coverage. Notably, only a minority of systems rely exclusively on header-level features (in fact 12 in their Table 1 were related to their column T0*), and even fewer are designed to function in the absence of data values.

5.3 Comparative Gaps and Trends

Table 5.1 Extension from Table 1 from Liu et al [23] is in appendix 8.3

Our Table 5.1 updates and expands Liu et al.'s Table 1 review with new systems, new benchmarks, and additional metadata-centric criteria. Several important trends emerge:

- **Limited Header-Only Systems**: Despite the rise of metadata-only tasks, few systems are truly capable of operating without cell values. Only 12 in Liu's study used T0* headers explicitly.

- **Static Dictionaries and Lack of Feedback**: Most lookup systems rely on static dictionaries, with little support for error-driven feedback or adaptive expansion.

- **Missing Integration with Quality Assessment**: No prior system combines semantic type detection with practical data quality evaluation, a key feature of our methodology.

5.4 Key Lessons from Liu et al.

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

- **Metadata is Underutilized**: While many systems rely on cell content, metadata such as headers offer rich and often underexploited signals.

- **Benchmark Bias Toward Single Ground Truth**: Benchmarks often penalize columns with multiple valid semantic types. Our approach accommodates ambiguity using NIL detection and multi-candidate scoring.

- **Knowledge Graph Sparsity**: KGs may lack appropriate matches for noisy or domain-specific headers. Our use of FinalFormat as an intermediate mapping layer helps overcome this limitation.

By aligning with Liu's framework and extending it through our new Table 5.1, we contribute to a deeper understanding of the trade-offs and challenges in modern CTA research.

6 CONCLUSION

We have presented a scalable and explainable header-centric framework for semantic type detection and data quality assessment. Unlike prior systems that rely heavily on cell values or deep models trained on static taxonomies, our approach leverages curated, feedback-enriched dictionaries and machine learning fallback to infer semantic types directly from column headers. This design enables our system to operate effectively in metadata-only settings, such as the SemTab 2024 challenge [12].

Our method demonstrated high coverage and accuracy across over 120,000 columns from diverse repositories, with only a modest increase in dictionary size. Through the integration of the *HeadersIQ* metric, we also introduced a novel, interpretable measure of column-level data quality—highlighting our framework's dual capacity for semantic enrichment and quality diagnosis.

By extending the taxonomy framework of Liu et al. [23] (see Section 5 for details), we demonstrated that our system advances current CTA paradigms, particularly in its ability to scale, generalize, and integrate with data curation pipelines. Future work will focus on refining ambiguity resolution, expanding multilingual support, and exploring fine-tuning strategies for large language models in header interpretation tasks.

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: 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; Wang-Chiew Tan. 2020. Sato: Contextual Semantic Type Detection in Tables. In Proceedings of KDD 2019, 1835–1848 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. 2017. 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