# APPENDICES

## Appendix A - Machine Learning Details (Attribute-BasedMLSemanticTypeDetection. ipynb)

This appendix documents the optional ML classifier pipeline, which complements the rule-based stage by producing per-column format predictions under header-only evidence. It details preprocessing, feature extraction, model training, and evaluation, and accompanies ***Algorithm 1***, which specifies where the ML classifier is invoked within the semantic type detection workflow.

For reproducibility, all configuration parameters; including random seed, data-split policy, cross-validation, hyper-parameter grids, and persistence of trained models; are listed at the end of this appendix.

***Scope note:*** In this study, the ML classifier was run only on the first **100 data sources (50 UCI + 50 Prague) with around 1,400 columns**; all other experiments use the **rule-based** pipeline alone.

**Preprocessing and Feature Engineering.**

Column headers and descriptions are cleaned through normalisation, abbreviation replacement, and camel-case splitting (see **Section 2.3**). The resulting textual data is vectorised using a TF–IDF model [33] built from a combined vocabulary of *DBHeaders* [9] and the *Formats dictionary*. In parallel, **dictionary-derived *SourceKeywords* are extracted to record which terms triggered candidate formats, supporting explainability, consistency checks, and enrichment when NIL cases are logged**.

**Addressing Class Imbalance.**

Semantic type classification often suffers from skewed distributions, with some types appearing far more frequently than others. To mitigate this imbalance, we apply the Synthetic Minority Over-sampling Technique (SMOTE), generating synthetic minority class samples to improve generalisation while reducing the risk of overfitting.

**Model Selection and Hyperparameter Optimisation.**

Multiple classifiers are evaluated, including Random Forest, Logistic Regression, Gradient Boosting, K-Nearest Neighbours, and LinearSVC. Hyperparameters are tuned via GridSearchCV with stratified cross-validation to ensure fair optimisation across imbalanced classes.

**Evaluation and Interpretability.**

Models are assessed using precision, recall, and F1-score on held-out test data. Feature importance analysis is employed to interpret classification behaviour and identify influential terms.

**Model Persistence.**

All trained classifiers, together with the fitted TF-IDF vectorizer, are saved to support reproducibility and allow flexible deployment scenarios.

**Algorithm 1 – Attribute-Based ML Semantic Type Detection**
*(applied systematically to all non-ID columns; dictionaries provide features and reconciliation)*

**Inputs**:
dataInfo (dataset/table metadata and columns); formatDict (Formats Dictionary); abbrDict (Abbreviations Dictionary); dbHeadersInfo (DB headers corpus); classifiers ({RandomForest, LogisticRegression, GradientBoosting, KNN, LinearSVC}); tfidfVectorizer (trained on Formats+DBHeaders vocabulary).

**Output**:
analysedColumns_models (per-column predictions per model + aggregate, confidence, NIL/flags, review logs).

```
function SemanticTypeDetect(dataInfo, formatDict, abbrDict, dbHeadersInfo,
        classifiers, tfidfVectorizer, confidenceThreshold)
  columns ← ExtractAndCleanColumns(dataInfo)    // trim, normalise case/punctuation, etc.

  for each column in columns do
    // Deterministic ID assignment (DB context)
    if column.belongsToDatabase then
      pkfk ← GetPrimaryForeignKeyInfo(column, dataInfo)
      if pkfk.isPK or pkfk.isFK then
        column.semanticType ← "ID column"
        column.mlConfidence ← 1.0
        SavePerModelPredictions(column, {})    // optional empty per-model record
        continue
      end if
    end if

    // Dictionary-derived FEATURES (no hard gate)
    expanded ← ReplaceAbbreviations(column.cleanedName, abbrDict)
    expanded ← SplitCamelCase(expanded)
    sourceKeywords ← ExtractKeywords(expanded, formatDict)    // e.g., ColumnKeyword, DescriptionKeyword

    // Compose text input and vectorise
    if column.description ≠ ∅ then
      inputText ← Concatenate(expanded, " ", column.description)
    else
      inputText ← expanded
    end if
    X ← tfidfVectorizer.transform([inputText])

    // Predict with all classifiers; collect per-model outputs
    preds ← {}    // model → (label, confidence)
    for each clf in classifiers do
      y_hat ← clf.predict(X)
```

```
    p_max ← PredictConfidence(clf, X, y_hat)    // calibrated prob / softmax / decision fn
    preds[clf.name] ← (y_hat, p_max)
  end for

  // Aggregate predictions (e.g., weighted by validation F1)
  (aggType, aggConf) ← AggregatePredictions(preds, weights=ValidationScores(classifiers))
  column.semanticType ← aggType
  column.mlConfidence ← aggConf

  // Reconciliation & review signals
  column.ruleSignals ← sourceKeywords    // retained for explainability
  column.consistencyFlag ← Consistency(ruleHint=BestRuleHint(sourceKeywords),
                                       mlType=aggType)    // AGREE / DISAGREE / NONE

  // Low-confidence handling → NIL + enrichment queue
  if aggConf < confidenceThreshold then
    column.semanticType ← "NIL"
    LogForReview(column, sourceKeywords, TopK(preds, k=3)) // store inputs + top candidates
    AppendToEnrichmentQueue(sourceKeywords, column)    // drive dictionary/abbr updates
  end if

  SavePerModelPredictions(column, preds)    // persist all models' outputs
 end for

 return analysedColumns_models(columns)
end function
```

**Reproducibility details.**

We fix the random seed to **42** across all steps. Data are split with **80/20 stratified hold-out** (train_test_split(test_size=0.2, stratify=labels)), and model selection uses **GridSearchCV** with **3-fold CV** (cv=3, n_jobs=-1). **Imbalance handling:** we apply **SMOTE** with k_neighbors=nn (as in the released code) and a task-specific sampling_strategy. **Feature extraction:** a custom **TF–IDF** vectorizer built from the **union** of **DBHeaders** tokens and the **Formats** dictionary (as normalised in §2.3).

**Evaluated models & grids:**

**RandomForest** (n_estimators ∈ {100,200}, max_depth ∈ {None,30}, min_samples_split ∈ {2,5}),
**LogisticRegression** (C ∈ {0.1,1}, solver='lbfgs', max_iter=20000),
**GradientBoosting** (n_estimators=100, learning_rate=0.1, max_depth=3),
**KNN** (n_neighbors ∈ {3,5}, weights ∈ {uniform,distance}),
**LinearSVC** (C ∈ {0.1,1}, max_iter=20000, dual='auto').

**Persistence:** the selected model per classifier and the TF–IDF vectorizer are exported via **joblib** to support exact reproduction.
**Repository:** code, dictionaries (Formats/Abbreviations), range tables, and the FinalFormat↔KG mapping tables are released in [38].

## Appendix B - Bounded Numerical types

This appendix lists the bounded numerical formats adopted throughout the rule-based and data-quality pipelines. Each of these formats defines a numeric attribute whose valid range is defined a priori, enabling automatic detection of out-of-range and physically impossible values during data quality assessment (***Section** 2.7*). The bounds shown below were derived from domain standards and dataset statistics and are applied uniformly across all evaluated data sources. These limits are **not fixed thresholds,** they are intentionally subjective and can be easily altered to reflect alternative domain assumptions or updated business rules, for example, different accepted ranges for *year* or *age*.

**Table A**

| 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 |

## Appendix C - TOP 100 list of NIL tokens ordered by amount

| # | CleanedColumn | Count | # | CleanedColumn | Count |
|---|---|---|---|---|---|
| 1 | no | 278 | 21 | sa | 30 |
| 2 | so | 149 | 22 | f | 30 |
| 3 | Nil | 98 | 23 | wp | 28 |
| 4 | min | 80 | 24 | gd | 26 |
| 5 | ga | 73 | 25 | st | 25 |
| 6 | sv | 73 | 26 | rc | 24 |
| 7 | + | 71 | 27 | off | 22 |
| 8 | to | 68 | 28 | s no | 21 |
| 9 | lg | 64 | 29 | pl | 21 |
| 10 | s | 60 | 30 | m | 21 |
| 11 | er | 56 | 31 | maf | 21 |
| 12 | gf | 49 | 32 | 2 | 21 |
| 13 | cg | 49 | 33 | 1 | 20 |
| 14 | sf | 45 | 34 | 1b | 19 |
| 15 | from | 40 | 35 | c | 19 |
| 16 | pp | 36 | 36 | ch | 19 |
| 17 | dp | 34 | 37 | zero | 19 |
| 18 | po | 33 | 38 | pb | 19 |
| 19 | b | 32 | 39 | yc | 18 |
| 20 | ba | 32 | 40 | tav | 18 |

| # | CleanedColumn | Count |
|---|---|---|
| 41 | shg | 18 |
| 42 | 3 | 17 |
| 43 | fc | 17 |
| 44 | bf | 17 |
| 45 | x | 17 |
| 46 | out | 17 |
| 47 | sl no | 16 |
| 48 | bs | 16 |
| 49 | new | 16 |
| 50 | lvl | 16 |
| 51 | streak | 16 |
| 52 | for | 15 |
| 53 | us | 15 |
| 54 | medal | 15 |
| 55 | otl | 15 |
| 56 | designation | 15 |
| 57 | lost | 14 |
| 58 | fs | 14 |
| 59 | vote | 14 |
| 60 | chest | 14 |
| 61 | gw | 14 |
| 62 | ta | 14 |
| 63 | extension | 14 |
| 64 | stats | 14 |
| 65 | ff | 14 |
| 66 | nat | 14 |
| 67 | tries | 13 |
| 68 | tdp | 13 |
| 69 | dep | 13 |
| 70 | defense | 13 |

| # | CleanedColumn | Count |
|---|---|---|
| 71 | replies | 13 |
| 72 | mat | 13 |
| 73 | prd | 13 |
| 74 | may | 13 |
| 75 | toi | 13 |
| 76 | rf | 13 |
| 77 | ot | 13 |
| 78 | ra | 13 |
| 79 | away | 13 |
| 80 | str | 12 |
| 81 | warp | 12 |
| 82 | generosity | 12 |
| 83 | stds | 12 |
| 84 | winnings | 12 |
| 85 | thailand | 12 |
| 86 | bpg | 12 |
| 87 | guild | 12 |
| 88 | arr | 12 |
| 89 | bop | 12 |
| 90 | wpa | 12 |
| 91 | pt | 12 |
| 92 | joined | 12 |
| 93 | realm | 11 |
| 94 | station | 11 |
| 95 | delta | 11 |
| 96 | @bat | 11 |
| 97 | previous | 11 |
| 98 | adult | 11 |
| 99 | subcategory | 11 |
| 100 | msrp | 11 |

## Appendix D - Preprocessing and Semantic Type Assignment: Detailed Workflow

Our semantic type detection pipeline employs a comprehensive preprocessing and mapping approach that transforms raw column headers into interpretable semantic annotations via the ***FinalFormat*** type system. This process involves a series of systematic steps, each designed to maximize robustness, flexibility, and explainability.
Our semantic type detection pipeline executes the following steps, closely mirroring the production implementation:

**1. Initial Parsing and Metadata Extraction:**

- Extract numeric prefixes (e.g., "1.ID") for traceability.
- Parse column names by splitting at colons, slashes, or parentheses to isolate core terms.
- Extract free-text descriptions from split points, or from parentheses if none exists, ensuring use of all available metadata.

**2.Header Normalization:**

- Convert all header tokens to lowercase.
- Remove or replace punctuation and special characters (underscores, hyphens, asterisks).
- Split compound words using spaces, dots, underscores, hyphens, or camelCase patterns (e.g., "AvgPitStopTime" → "avg pit stop time").
- Explicitly tokenize symbols (e.g., "%") for percentage detection.
- Match plural and singular forms (e.g., "chlorides" → "chloride").

**3. Abbreviation and Variant Expansion**

- Replace abbreviations using a dictionary of over 1000 entries (e.g., DOB → date of birth; bp → blood pressure).
- Apply regular expressions to tokenize and substitute complex patterns, including units (mg/dL, kg/m$^2$), and non-alphabetic tokens like ºC, cm$^2$, etc.

**4. SourceKeywords Extraction**

- Identify key tokens (SourceKeywords) that capture semantic meaning (e.g., "age", "date", "amount").
- Scan descriptions for known tokens, including "has", "is", or numeric/temporal keywords.
- Use primary/foreign key metadata to override assignments where appropriate.

**5. Semantic Type Assignment (Rule-Based and ML classifier Logic)**

- Match tokens to a curated dictionary (2,800 mappings) to assign one of 39 FinalFormat types.
- Apply priority logic:
  - "binary" overrides all if "has", "is", or "or not" is present.
  - "percentage" takes precedence if "%" or related terms detected.
  - Hierarchical rules resolve conflicts between categorical, numerical, and domain types.
  - Special logic for "name" columns in certain contexts (city, state, etc.).
- If no match is found:
  - Retry abbreviation expansion and check individual tokens (with plural handling).
  - If still unresolved, use ML-based prediction.

**6. Ambiguity Resolution**

- Use ML classifier (Random Forest, Logistic Regression, etc.) with weighted TF-IDF features.
- Assign confidence scores to all assignments for human-in-the-loop review.

**7. Knowledge Graph Annotation**

- Perform syntactic and semantic matching (FastText) between normalized headers and KG property/class labels.
- Use substring and similarity logic for best match selection (DBpedia, Schema.org).

**8. Final Type Selection and Logging**

- Consolidate assignments if column and description differ, prioritizing by semantic importance (e.g., prefer “datetime”, “IDcolumn”, etc.).
- Attribute assignments to specific source keywords for explainability.
- Assign “Nil” and log cases for manual review if no match is found.

Our semantic type detection pipeline applies a comprehensive sequence of normalization, expansion, and mapping steps that transform raw column headers into interpretable semantic annotations via the FinalFormat type system. The workflow (detailed above) enables robust, transparent, and flexible type assignment across real-world tabular data.

**Practical Examples of Semantic Type Assignment**

***Table B*** below demonstrates, on a large and diverse set of real columns, how our pipeline’s dictionary logic, abbreviation expansion, and contextual rules deliver precise, explainable semantic types.

**Key Interpretation Patterns and Highlights**

**Binary Columns:**

Binary-type columns are often signaled by common words such as “has”, “is”, or phrases like “or not”. The pipeline detects these cues and classifies such columns under the binary FinalFormat, ensuring boolean attributes are annotated correctly even when explicit True/False values are absent.

**Disambiguating Categorical Types:**

Columns containing the word “range” (e.g., “price range”, “date range”) are accurately identified as categorical, despite potentially misleading co-occurring terms like “date” or “price” that might otherwise point toward temporal or financial types. This illustrates the nuanced, context-aware mapping the system achieves via dictionary-based disambiguation.

**Geographic and Demographic Mapping:**

Headers with words like “nationality”, “province”, or “capital” are consistently mapped to country, state, or city FinalFormats. Synonym handling ensures that “ProvinceName” is linked to state, and “capital name” to city, even if these terms don’t match dictionary entries exactly.

**Temporal and Timestamp Recognition:**

Fields such as “updated”, “day of week”, and “timestamp” are mapped to their appropriate FinalFormats:
“updated” → date
“request timestamp” → datetime
“day of week” → weekday
This allows precise differentiation between temporal and generic date columns.

**Financial and Monetary Attributes:**

Lexical cues like “cash”, “currency”, “usd”, and “price” reliably trigger the assignment to the money FinalFormat, enabling wide coverage of financial information.

**Communication and Identifier Columns:**

Terms like “fax”, “cellular phone”, and “author channel id” are normalized to phone and IDcolumn, respectively, with common abbreviations (e.g., “nr” for number) mapped to numerical.

**Scientific and Measurement Values:**

Abbreviations and specialized terms, such as “wgt avg” (weight average) or “trestbps” (blood pressure), are interpreted using expansion rules, with assignments to numerical>=0 and bloodpressure, reflecting the system’s capacity to handle both generic and domain-specific semantics.

**Web and Address Columns:**

Patterns like “web address”, “link”, or “href” result in URLformat annotations, unifying various forms of URL-related fields.
Postal and address information, including “zipcode”, “provider zip code”, and “purchase address”, is reliably mapped to postalcode or street.

**String and Free-Text Columns:**

Terms such as “abstract”, “desc 1”, and “spec abstract” are interpreted as free text and assigned the string FinalFormat.

**Other Notable Patterns:**

- Medical/Health: Medical headers like “bp” or “trestbps” are mapped to bloodpressure.
- Percentage: Any header containing the “%” symbol or “percent” is assigned percentage.
- Temporal granularity: “Month”, “hour”, “week number”, “year” are all mapped to their corresponding temporal FinalFormats.

Through these patterns, ***Table B*** provides a transparent view of our pipeline’s ability to handle abbreviation, synonymy, compound phrases, and real-world header messiness, delivering precise, actionable semantic types for data quality assessment.

**Table B**

| Original Header | Normalized | 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** |
| BookedHotelOrNot | booked hotel **or not** | **or not** | **binary** |
| bp | **bp** | **blood pressure** | bloodpressure |
| trestbps | **trestbps** | **blood pressure** | bloodpressure |
| Reported Influenza Activity | reported influenza activity | activity | categorical |
| Aircraft Manufacturer | aircraft manufacturer | aircraft | categorical |
| 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** |
| CapitalName | **capital** name | **capital** | **city** |
| Author, Country | author, **country** | country | country |
| team_one_player_one_nationality | team one player one **nationality** | **nationality** | **country** |
| BIRTHDATE | **birthdate** | birthdate | date |
| dtRef | **dt** ref | date | date |
| Data Last Updated | data last **updated** | **updated** | **date** |
| Order Date and Time | order **date and time** | date and time | datetime |
| Request timestamp | request **timestamp** | **timestamp** | **datetime** |
| school_day | school **day** | day | day |
| FLAG_EMAIL | flag **email** | email | E-mailformat |
| order_hour_of_day | order **hour** of day | hour | hour |
| authorChannelId | author channel **id** | id | IDcolumn |
| Nameserver IP Address | nameserver **ip** address | ip | IPformat |
| src_ip_country_code | src **ip** country code | ip | IPformat |
| vehicle_gps_latitude | vehicle gps **latitude** | latitude | latitude |
| Delivery_location_longitude | delivery location **longitude** | longitude | longitude |
| Cash amount | **cash** amount | **cash** | **money** |
| discount_price__currency | discount price **currency** | **currency** | **money** |
| cons.price.idx | cons **price** idx | **price** | **money** |
| Values in Billions USD | values in billions **usd** | **usd** | **money** |
| Date_Of_Death_Month | date of death **month** | month | month |
| Head Coach | head **coach** | **coach** | **name** |
| Artistic Director | artistic **director** | **director** | **name** |
| fullname words | **fullname** words | **fullname** | **name** |
| LASTNAME | **lastname** | lastname | name |
| Person Baptised | **person** baptised | person | name |
| Allowed Amount | allowed amount | **amount** | **numerical** |
| Trip_Distance_km | trip distance km | **distance** | **numerical** |
| Ground Elevation | ground **elevation** | **elevation** | **numerical** |
| Flug-Nr. | flug **nr** | **number** | **numerical** |
| Deforestation_Area_Ha | deforestation **area** ha | **area** | **numerical>=0** |
| free_throw_attempts | free throw **attempts** | **attempts** | **numerical>=0** |
| ViewDepth | view depth | **depth** | **numerical>=0** |
| DirectoryEntryImportSize | directory entry import **size** | **size** | **numerical>=0** |
| Wgt. Avg. | **wgt avg** | **weight** | **numerical>=0** |
| YearsInCurrentRole | **years** in current role | **years** | **numerical>=0** |
| % of Total Supply Owned | **%** of total supply owned | **%** | **percentage** |
| Percent of State employment | **percent** of state employment | **percent** | **percentage** |
| Number of Cellular phone | number of **cellular phone** | cellular phone | phone |
| Fax Numbers | **fax** numbers | **fax** | **phone** |
| Customer_Postal_Code | customer **postal code** | postal code | postalcode |
| Provider Zip Code | provider **zip code** | **zip code** | **postalcode** |
| ProvinceName | **province** name | **province** | **state** |
| Purchase Address | purchase **address** | address | **street** |
| Spec. abstract | spec **abstract** | abstract | string |
| desc_1 | **desc** 1 | **description** | **string** |
| AvgPitStopTime | avg pit stop **time** | time | time |
| goods-title-link--jump href | goods title link jump **href** | **href** | **URLformat** |
| Free Download Link | free download **link** | **link** | **URLformat** |
| Web Address | **web address** | **web address** | **URLformat** |
| arrival_date_week_number | arrival date **week** number | week number | week |
| Day of Week | day of week | **day of week** | **weekday** |
| athlete_year_birth | athlete **year** birth | year | year |

These examples illustrate how the system transforms raw, often noisy or abbreviated, column headers into precise semantic annotations. By combining normalization, robust abbreviation expansion, and hierarchical rule logic, our pipeline ensures reliable type assignment even in the face of ambiguity, domain-specific terminology, or non-standard header formats. This approach not only supports accurate data profiling but also underpins the effectiveness of our automated data quality assessment framework. Columns that remain unresolved after this rule-based process are passed to the **machine-learning classifier** described in ***Appendix A***, which provides detailed configuration and reproducibility information for that stage.

## Appendix E - Full SemTab 2024 Metadata-to-KG Track [12] Analysis

### E.1 Methods (Human Review Protocol)

One author, **blinded to official GT labels and to all cell values**, adjudicated each prediction using only the **column header** and public ontology documentation (definitions and official redirects). Each case was assigned to one of four categories:

- **C1 (GT-CONF):** Predicted URI equals the GT URI or an **official redirect to the same canonical entity**. *(None were found beyond the official script; C1 contributes 0 in our audit.)*
- **C2 (GT-REFINE):** Predicted URI is **not** the GT entity but is a **more appropriate** entity/property for the header (change of granularity/scope/type that better reflects the header). We submit these as **candidate GT issues**.
- **C3 (ALT):** Predicted URI is **not** the GT entity and **not necessarily better**, but remains **plausible** under header-only interpretation (synonym/alias/generalisation).
- **C4 (INC):** Remaining cases (NIL, non-English unresolved, unit/numeric-only, ambiguous, lexical confusion, etc.).

We report raw counts and, where space permits, **95% binomial Wilson confidence intervals** for proportions.

### E.2 Results Overview (n = 141)

**Official (GT-strict).** Using exact URI equality for the same canonical entity (no manual flags):

- **Hit@1 (strict): 63/141 ≈ 0.45** (95% CI ≈ **[0.36, 0.53]**)
- **Hit@5 (strict): 67/141 ≈ 0.48** (95% CI ≈ **[0.40, 0.56]**)

**Audit reclassification (ours only; header-only rubric).** From the adjudicated set:

- **C1 (GT-CONF, additional): 0** *(no extra same-entity corrections beyond strict)*
- **C2 (GT-REFINE): 43/141 (30.5%)**
- **C3 (ALT): 6/141 (4.0%)**
- **C4 (INC): 24/141 (17.0%)**

**Diagnostic Hit@1 (ours only; not for cross-system ranking).**

- **B1: Strict: 63/141 = 0.447** (≈0.45)
- **B2: B1 + C2 (GT-REFINE): (63+43)/141 = 106/141 = 0.752** (95% CI ≈ **[0.67, 0.81]**)
- **B3: B2 + C3 (ALT): (106+6)/141 = 112/141 = 0.794** (95% CI ≈ **[0.72, 0.85]**)

**Diagnostic Hit@5 (ours only; header-only audit)**

Starting from the strict Top-5 match count (**67/141 = 0.475**), we apply the same cumulative reconciliation used for Hit@1:

- **B1 — Strict: 67/141 = 0.475** (95% CI ≈ **[0.40, 0.56]**)
- **B2 — B1 + C2 (GT-REFINE): (67 + 43)/141 = 110/141 = 0.781** (95% CI ≈ **[0.71, 0.85]**)
- **B3 — B2 + C3 (ALT): (110 + 6)/141 = 116/141 = 0.823** (95% CI ≈ **[0.76, 0.88]**)

**Notes.**

1. **C1 (GT-CONF)** contributed **0** cases; therefore Panel B starts from the strict baseline.
2. These figures are **diagnostic** and apply to **our outputs only**; they are **not used for cross-system ranking**.
3. Under this diagnostic view, **C2/C3 are counted as correct** for Top-5 irrespective of whether the GT URI appears in the model's Top-5 list, because the human audit judged them ontology-consistent and header-appropriate under the rubric.

### E.3 Tables

**Table C - GT-Refinement (Author-Proposed Replacement)**

| Column values | GT Annotation | Our Annotation | Items | Justification |
|---|---|---|---|---|
| Year | dbo:releaseDate | dbo:year | 20 | **Granularity**: header denotes general temporal attribute; releaseDate presupposes an event/product. |
| ISO(2)/ISO(3) | dbo:iso31661Code | dbo:isoCode | 2 | **Scope**: header indicates generic ISO code; 3166-1 scope not evidenced by header alone. |
| Length | dbo:duration | dbo:length | 1 | **Type**: physical extent vs time interval. |
| Government | dbo:governmentType | dbo:government | 4 | **Entity vs attribute**. |
| State | dbo:location | dbo:state | 4 | **Geopolitical specificity**. |
| GDP | dbo:grossDomesticProduct | dbo:**gdp**PerCapita | 1 | **Indicator specificity** (flag as refinement with caveat). |
| Alphabetic**code** | dbo:currencyCode | dbo:code | 4 | **Generalisation**: currency scope not evidenced. |
| Granting**Institution** | dbo:almaMater | dbo:institution | 1 | **Relation vs institution**. |
| **Height**(ft), **HEIGHT**INMETERS | dbo:elevation | dbo:height | 5 | **Domain shift**: non-geographical height likely. |
| Board | dbo:owner | dbo:board | 1 | **Role mismatch**. |
| | | **TOTAL** | **43** | **30.5%** |

**Table D - Ontology-Consistent Alternative (Plausible, not GT)**

| Column values | GT Annotation | Our Annotation | Items | Justification |
|---|---|---|---|---|
| Size | dbo fileSize | dbo collection**Size** | 1 | Header ambiguity; plausible alternative. |
| Label | dbo developer | dbo distributing**Label** | 1 | Label semantics vs developer. |
| System | dbo computingPlatform | dbo **system**Requirements | 3 | Ambiguous "System"; both plausible. |
| **water**gauge | dbo elevation | dbo water | 1 | Water-level semantics plausible. |
| | | **TOTAL** | **6** | **4%** |

**Table E - Incorrect (incl. NIL / multilingual / unit-only / lexical confusions)**

| Column values | GT Annotation | Our Annotation | Items | Issue |
|---|---|---|---|---|
| Basincountries, Staat(en) | country | Nil | 2 | Multilingual/compound words issues |
| 8.848, Feets, Meters, H?he | elevation | Nil | 5 | Numeric/unit headers lacking context |
| GNI, GDPnominal(US$M) | giniCoefficient, grossDomesticProduct | Nil | 2 | Abbreviations and economic indicators |
| Ashton-under-Lyne, Editeur, Endroit, Stadt | location, owner | Nil | 5 | Non-English terms, ambiguous without context |
| Europe, Japan, Rel. | releaseDate | Nil | 5 | Contextual mismatch with GT annotation |
| NorthAmerica | releaseDate | northWestPlace | 1 | Contextual mismatch with GT annotation |
| Rationale | knownFor | ratio | 2 | Semantic misunderstanding due to lexical similarity |
| 1T | iataAirlineCode | Nil | 1 | headers lacking context |
| Max.-Tiefe-(m) | depth | max | 1 | headers lacking context |
| | | **TOTAL** | **24** | **17%** |

**E.4 Successes and Challenges (Header-Only CTA)**

**Scope.** This section interprets results under the **header-only** constraint and uses the audit rubric from §E.1: **C.1 GT-Confirmed** (same canonical entity), **C.2 GT-Refinement** (author-proposed replacement for GT), **C.3 Ontology-Consistent Alternative** (plausible but not GT), and **C.4 Incorrect** (incl. NIL/ambiguous/multilingual/unit-only).

**E.4.1 Success cases (C.1 / C.3)**

A substantial fraction of columns with clear headers are annotated reliably:

- **Examples (C.1 GT-Confirmed where applicable).** Headers such as "Government", "Year", and "Height(ft)" tend to map to intuitive DBpedia targets. When the **same canonical URI** as GT is reached (identical URI or official redirect), these are counted under **C.1**.
- **Plausible alternatives (C.3).** In some cases the model selects ontology-consistent alternatives that remain defensible under header-only semantics, e.g., elevation → height for non-geographic contexts, developer → distributingLabel for "Label", or computingPlatform → systemRequirements for "System". These are recorded as **C.3** because they are plausible yet **not** the GT entity.

Takeaway. Under the header-only premise, the system is robust to minor label differences and often converges to semantically sound choices even when GT uses a different—but related—entity/property.

**E.4.2 Error and challenge cases (C.4)**

Most failures arise from classes of headers that are inherently fragile under header-only interpretation:

- **Multilingual/compound or abbreviated headers.** E.g., "Basincountries", "Staat(en)", "Editeur", "Rel."
- **Unit-only or numeric-only headers.** E.g., "8.848", "Feets", "Meters", "Max.-Tiefe-(m)".
- **Code/identifier without scope.** E.g., "1T" (IATA airline code).
- **Ambiguous geography vs. temporal GT.** E.g., "Europe", "Japan", "NorthAmerica" marked as releaseDate in GT; such mappings appear to rely on **cell values**, not headers.

We record these under **C.4 Incorrect** (including **NIL** when the system declines to guess). In header-only evaluation, **NIL** can be desirable behaviour for high-precision operation.

**E.5 Human-Informed Diagnostic of GT Inconsistencies (C.2 / C.3)**

**Motivation.** Strict GT requires the predicted URI to match the **same canonical entity** as the ground truth. However, benchmarks may encode **granularity** or **context** that is not present in column headers. We therefore performed a blinded, header-only audit to separate **(i)** genuine model errors from **(ii)** GT idiosyncrasies.

**E.5.1 GT-Refinement (C.2 author-proposed replacements)**

We identified cases where the model's URI appears **more appropriate** for the header than the GT entity (candidate GT issues). Representative examples:

- **Year:** releaseDate → year (n=20). The header denotes a general temporal attribute; releaseDate presupposes an event/product.
- **ISO(2)/ISO(3):** iso31661Code → isoCode (n=2). The header does not evidence the 3166-1 scope; a scope-neutral ISO code is header-faithful.
- **Length:** duration → length (n=1). Physical extent vs. time interval.
- **Government:** governmentType → government (n=4). Institution vs. classification attribute.
- **State:** location → state (n=4). Geopolitical specificity.
- **Alphabeticcode:** currencyCode → code (n=4). General "code" without currency scope in the header.
- **GrantingInstitution:** almaMater → institution (n=1). Granting body vs. alumni relation.
- **Height(ft), HEIGHTINMETERS:** elevation → height (n=5). Non-geographical height signalled by header.
- **Board:** owner → board (n=1). Governance/advisory body vs. ownership.

Classification. All such cases are recorded as **C.2 GT-Refinement** (not counted as "Correct" under strict GT), and fed into **Panel B** diagnostics to quantify potential GT noise in header-only settings.

**E.5.2 Ontology-Consistent Alternatives (C.3)**

We also found plausible alternatives that are defendable under header-only semantics, but without claiming they "correct" GT. Examples include "Size: fileSize → collectionSize", "System: computingPlatform → systemRequirements", and "watergauge: elevation → water". These are recorded as **C.3**.

Diagnostic use only. Neither **C.2** nor **C.3** is used for cross-system ranking; they solely inform **Panel B** (ours only) to show how strict scores shift when GT granularity/aliasing is accounted for.

**E.6 Consolidated Implications (Header-Only CTA)**

- **Strict baseline preserved.** All comparative conclusions in the paper rely on **official GT-strict** results (Panel A).
- **Quantified GT noise.** The audit suggests that ~**30.5%** of residual strict-GT "errors" reflect **GT-Refinement** (C.2) and ~**4%** reflect **Plausible Alternatives** (C.3) under header-only interpretation, lifting diagnostic Hit@1 from **0.45** to **≈0.76** (E.2) and **≈0.80** (C.2+C.3).
- **Benchmark design.** We recommend that future benchmarks distinguish **header-only** targets from **context-dependent** targets and maintain synonym/alias sets per entity/property to reduce avoidable mismatches.

**E.7 Researcher-Proofing and Artifacts**

- **Uncertainty reporting.** For our measures, we report **raw counts** and **95% Wilson CIs**; baseline CIs are omitted if not provided in the official report.
- **Reproducibility files.** We release a XLSX with one row per column: SEMTAB_2024_dbpedia_match_results.xlsx [38]
- **Namespace coverage.** Some DBpedia concepts exist only as capitalised **classes** (e.g., dbo:Airport) without a lowercase property; systems restricted to properties may miss such cases. We index **both class and property** namespaces to mitigate false negatives.
- **Task alignment.** CTA is a **column-level** assignment task; benchmarks like **t2Dv2** provide **table-level** classes and are not directly comparable to column-level CTA outcomes.

## Appendix F - Comparative Analysis of Semantic Type Detection Approaches

This appendix presents an **extended** comparative analysis of representative systems for semantic type detection. ***Table F*** consolidates the survey of Liu et al. [23] and **updates/extends** it with recent approaches (including **LLM-based metadata** methods) and **this header-centric framework**.
Dimensions covered include system class, header usage, **header-only capability**, knowledge-graph integration (DBpedia/Schema.org), benchmark datasets, **SemTab setting/participation** (including **offline comparisons**), and key techniques; offering a concise view of recent trends and gaps.

**Table F - Extended comparative analysis of semantic type detection approaches, adapted and expanded from Table 1 in Liu et al. [23].**

| System (Algorithm) | Year | Class | Header Usage | Header-only Capable? | KG | Data Source | SemTab? | Key Technique |
|---|---|---|---|---|---|---|---|---|
| Wang et al. [44] | 2012 | Heuristic, Lookup | Primary | No | Probase | Custom Wikipedia Tables | Baseline | Probase + rules |
| $C^2$ [19] | 2021 | Heuristic, Lookup | Probabilistic | No | DBpedia, Wikidata | Limaye, ISWC2017, SemTab 2019, T2D | Winner CTA 2020 | Ensemble/statistics |
| Magic [40] | 2021 | Heuristic, Lookup | Direct compare | No | DBpedia, Wikidata | SemTab 2021 | Participant | INK embeddings |
| Alobaid et al. [3] | 2022 | Heuristic, Lookup | Main input | Yes | DBpedia | SemTab 2021, T2D | Participant | String similarity & normalization |
| TableMiner+ [47] | 2017 | Heuristic, Iterative | Lexical | No | Freebase | Limaye, IMDB, MusicBrainz | Baseline | Lexical/iterative |
| CSV2KG [41] | 2019 | Heuristic, Iterative | Fallback | No | DBpedia | SemTab 2019 | Baseline | Heuristic |
| Mtab [29, 30, 31] | 2019 | Heuristic, Iterative | Ensemble | No | DBpedia, Wikidata | SemTab 2019–2021 | Winner | Lookup, NLP ensemble |
| Limaye et al. [22] | 2010 | Feature Engineering | Strong | No | YAGO | Limaye | Baseline | Prob. graphical model |
| Mulwad et al. [27,28] | 2010 | Feature Engineering | Limited | No | Wikitology | Limaye | Baseline | Heuristic, SVM |
| DAGOBAH Embeddings [6] | 2019 | Deep Learning, KG Modelling | Central | No | DBpedia, Wikidata | SemTab 2019 | Baseline (CTA) | ML+heuristic ensemble |
| Sherlock [14] | 2019 | Deep Learning, Table Modelling | Minimal (opt.) | No | DBpedia | T2D, VizNet | Baseline | Deep CNN |
| Sato [45] | 2020 | Deep Learning, Table Modelling | No header | No | DBpedia | VizNet | Baseline | TabNet/CRF/BERT |
| TURL [10] | 2020 | Deep Learning, Table Modelling | Limited | No | DBpedia | WikiGS, WikiTable, T2D | No | Tabular transformer |
| Doduo/BERT-CTA [42] | 2022 | Deep Learning, Table Modelling | Limited | No | - | WikiTable, VizNet | Participant | Fine-tuned BERT (multi-task) |
| RECA Sun et al [39] | 2023 | Deep Learning, Table Modelling | Limited | No | DBpedia | T2Dv2, SemTab, Wikipedia | No | Related table context + features |
| ADWAN | 2024 | LLM-based, Metadata | Prompts LLM | Yes | Dbpedia, Schema.org | SemTab2024 Metadata Only track | Winner | RAG + CoT + Self-Consistency + RRF |
| CVA/Metalinker(2024) | 2024 | LLM-based, Metadata | Main input | Yes | Dbpedia, Schema.org | SemTab2024 Metadata Only track | Participant | Zero-shot LLMs + RAG + SemanticBERT |
| Anonymized [37] | 2024 | Heuristic, Lookup | Exclusive header | Yes | - | UCI | No | Dynamic, feedback-driven, hybrid fallback |
| This paper | 2026 | Heuristic, Lookup | Exclusive header dynamic | Yes | Dbpedia, Schema.org | Kaggle, VizNet, Sato, UCI, Prague, SemTab 2024 Metadata Only track | 2024 Offline comparison | Rule-based + feedback/ML fallback |

*Notes:* (i) "Header-only" indicates operation without cell values; (ii) "SemTab note" distinguishes official participation from **offline** evaluations; (iii) rows and sources were cross-checked against cited papers; the curation criteria and change log are available in [38] for reproducibility.