International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.63282/3050-9246.IJETCSIT-V7I1P121
Eureka Vision Publication | Volume 7, Issue 1, 146-149, 2026

Original Article

Automated Data Lake Onboarding: A Hybrid Metadata and Content-Based Approach for Schema Matching

Sai Prashanth Pathi
Independent Researcher, USA.

Received On: 26/12/2025 ***Revised On: 25/01/2026*** ***Accepted On: 04/02/2026*** ***Published On: 12/02/2026***

Abstract - In large-scale enterprise environments, onboarding disparate asset teams to a centralized Customer Data Lake (CDL) or similar enterprise data repository is often bottlenecked by manual Gap Analysis. This process involves comparing legacy asset tables with the central lake to identify schema overlaps and missing information, a task complicated by non-standardized column naming and vast data volumes. This paper proposes an automated framework for schema matching that utilizes a hybrid approach. We employ a metadata module using Jaro-Winkler string similarity with a domain-specific abbreviation dictionary, alongside a data content module utilizing K-Nearest Neighbors (KNN) classification on feature-engineered embeddings. A unique feedback loop allows the data model to iteratively improve the metadata dictionary. Experimental results demonstrate an F1 score of approximately 90.7%, significantly reducing manual mapping efforts and streamlining the onboarding process.

Keywords - Machine Learning, Schema Matching, Data Lake, Automated Onboarding, K-Nearest Neighbors, Metadata Management, Gap Analysis, Enterprise Information Integration.

1. Introduction

The consolidation of data into a centralized Customer Data Lake (CDL) is a critical step for organizations aiming to perform unified customer analysis. However, the "onboarding" process where distinct asset teams migrate or map their local datasets to the CDL presents significant challenges. A primary hurdle is the "Gap Analysis," a procedure required to determine how well the CDL satisfies the requirements of the onboarding asset tables. While our case study focuses on a Customer Data Lake in a commercial setting, the proposed framework is applicable to other relational data integration scenarios with heterogeneous schemas (for example, data warehouses and domain data marts).

Traditionally, gap analysis is a manual process where domain experts compare asset tables with CDL tables to identify present and missing information. This approach is resource-intensive, time-consuming, and prone to human error, particularly given that column naming conventions are rarely standardized across different departments.

To address this, we developed a system that automatically generates a mapping sheet and scorecard, linking asset (source) columns to potential CDL (target) columns with a calculated match score. This paper details the system's design, which operates under constraints of minimal computational resources and limited labeled data. We present a hybrid model combining linguistic (metadata) analysis and instance-based (data content) learning to achieve high-accuracy schema matching.

2. Related Work

Schema matching is a fundamental problem in database management, defined as the process of identifying semantic correspondences between elements of two schemas [1]. Approaches are generally categorized into schema-level (metadata-based) and instance-level (content-based) matching.

2.1. Metadata-Based Matching

Metadata approaches rely on schema information such as column names and data types. String similarity metrics, such as Levenshtein distance or Jaro-Winkler distance, are commonly used to match column headers [2]. However, these methods struggle when abbreviations or synonyms are used (e.g., matching *txn* to *order*) without an external dictionary or ontology [3]. While effective for standardized environments, metadata matching often fails in "data swamps" where documentation is sparse [4].

2.2. Instance-Based Matching

When metadata is opaque or non-standardized, instance-based matching analyzes the actual data content. Machine learning classifiers, including K-Nearest Neighbors (KNN) and decision trees, have been employed to classify columns based on data patterns [5], [6]. Our work extends this by employing character-level embeddings and pattern extraction to treat the matching problem as a multi-class classification task.

2.3. Hybrid Approaches

Recent literature suggests that combining metadata and instance-level matchers often yields better results than either alone [7]. Tools like "Data Tamer" have pioneered the use of machine learning to suggest mappings which are then verified by humans [8]. Our proposed system aligns with this trend by implementing a scoring mechanism that aggregates confidence from both linguistic similarity and data pattern recognition, specifically tailored for relational (RDBMS-style) data in a commercial environment.

The proposed system utilizes a dual-approach architecture: a Metadata Model and a Data Model, which feed into a unified scoring system.

3. Methodology

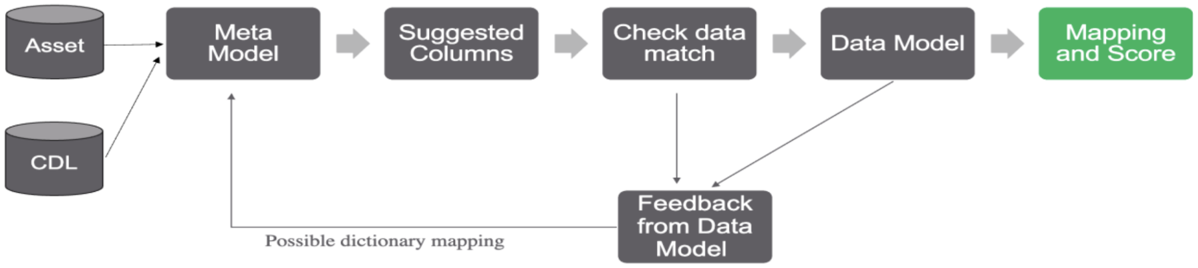

Figure 1. Metadata-Driven Data Mapping and Scoring Framework with Feedback Loop

Figure 1: Proposed system architecture. The system ingests data from asset and CDL sources, processing metadata via string similarity and data content via KNN classification. A feedback loop allows the Data Model to update the abbreviation dictionary used by the Metadata Model.

3.1. Metadata Module

The metadata module focuses on column names extracted from both the asset (source) and CDL (target) tables. 1) String Similarity: We utilize the Jaro-Winkler distance metric to calculate similarity scores between column name pairs. A threshold of 0.8 is established; pairs exceeding this score are flagged as potential mappings. 2) Abbreviation dictionary: Due to non-standardized naming (e.g., cust vs. customer), raw string similarity is often insufficient. We integrated a domain dictionary containing common English abbreviations (e.g., ts: timestamp) and domain-specific mappings (e.g., txn: order, d_lvl: division) to standardize names before comparison.

3.2. Data Module (KNN Classification)

The data module addresses scenarios where column names are ambiguous or missing. We formulate the problem as a multi-class classification task where each column in the baseline CDL table is treated as a separate class. We utilize a K-Nearest Neighbors (KNN) model trained on the CDL ecosystem tables.

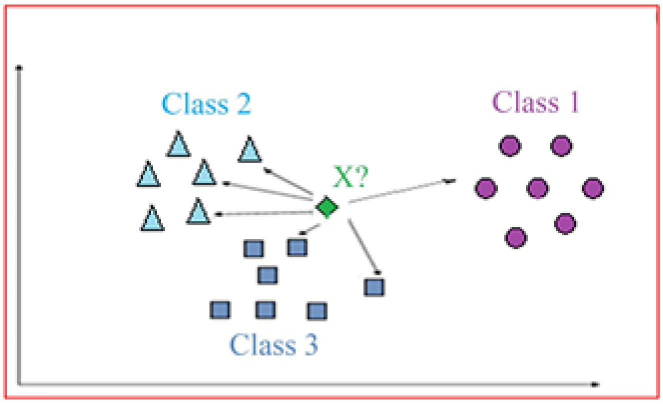

Figure 2. Multi-Class Classification Problem with Unknown Sample (X) Assignment

Figure 2: Visualization of the K-Nearest Neighbors (KNN) classification approach. Each column in the CDL baseline is treated as a distinct class. An unmapped asset column (*X*) is classified based on its proximity to these established classes using feature embeddings (e.g., character counts, data types).

Feature Engineering*:* Since raw data values (strings, dates, floats) cannot be processed directly by the model, we employ embedding and encoding techniques to capture underlying patterns:

- Character Embedding: Data values are treated as strings. Characters are mapped to numerical categories (e.g., Alphabet: 1, Digit: 8, Special Character: 3). For example, 2022-01-31 becomes 8888588588.
- Data Type Encoding: The SQL data type is encoded ordinally (e.g., string: 1, int: 3) to preserve type information.
- Pattern Statistics: We calculate counts of specific characters (alphabets, digits, dots, dashes) to generate a feature vector. For instance, ABCD.COM is represented as [6, 0, 1, 0, 0].

3.3. Feedback Loop

A novel component of this system is the automated feedback loop. The metadata model relies heavily on the quality of the abbreviation dictionary. When the metadata model fails to predict a match (or predicts one with low data similarity) and the Data Model predicts a match with high confidence (similarity score > 0.8), this new mapping is suggested as an addition to the dictionary.

- Example: The metadata model may fail to link category to purchase_superdept. If the data model identifies the relationship based on content, category is mapped to purchase_superdept and is added to the feedback loop.

3.4. Scoring Mechanism

The final mapping decision is based on a composite score for each column, derived from the maximum of the Meta Score and Data Score. The score for the table is the average of the individual column scores on a 0-10 scale.

- Meta_Score: Sum of 0.5 (if prediction exists) and 0.5 (if data types align).
- Data_Score: Sum of 0.2 (if model finds a match) and 0.8 (if predicted column name has high string similarity).

$$\text{Column\_score} = \max(\text{Meta\_Score}, \text{Data\_Score})$$
$$\text{Table\_score} = \text{Avg}(\text{Column\_scores})$$

4. Experimental Results

The system was tested using sample datasets characterized by RDBMS-style structure. The primary evaluation metric was the F1 score.

4.1. Performance

The model achieved an F1 score of 90.7% (excluding indicator columns). The system successfully quantified similarity between asset and CDL tables. For example, the *cnsld_order_item_trans* asset table showed a 9.1 similarity score with the CDL *cust_scan_catalog* table, while *interactions_catalog* received a similarity score of 6.1. Table 1 shows how the asset table *cnsld_order_item_trans* is evaluated against three CDL tables. The resulting similarity scores indicate that *cust_scan_catalog* is the optimal mapping table, achieving the highest score of 9.1.

4.2. Mapping Accuracy

The system correctly identified complex mappings such as *op_cmpny_cd* (value: *"ABCD.COM"*) mapping to *cmp_cd* and *order_plcd_ts* mapping to *order_tmstp*. Table 2 illustrates the comparative logic used during result validation.

Table 1. Sample Table Mapping Results

Asset Table	CDL Tables	Similarity Score
cnsld_order_item_trans	customer_transaction	7.3
	cust_scan_catalog	9.1
	interactions_catalog	6.1

Table 2. Sample Column Mapping Results

Asset Column	Meta Prediction	Data Check	Data Model Prediction	Result
order_nbr	txn_nbr	TRUE	txn_id	Match
order_item_id	order_item_id	FALSE	svc_id	Mismatch/Flagged

5. Conclusion and Future Work

This study presented a novel, automated framework for streamlining the "onboarding" process of disparate asset teams into a centralized Customer Data Lake (CDL). By addressing the critical bottleneck of manual "Gap Analysis," which traditionally requires extensive human intervention to map non-standardized schemas, we demonstrated that automation is both feasible and highly effective in enterprise environments.

5.1. Summary of Contributions

The core contribution of this work is the development of a hybrid schema matching architecture that functions under the constraints of limited labeled data and limited resources.

- Hybrid Efficacy: We demonstrated that relying solely on metadata is insufficient due to inconsistent naming conventions. By augmenting the metadata approach with a content-based K-Nearest Neighbors (KNN) model utilizing character embeddings and pattern statistics we achieved a robust F1 score of 90.7%.
- Adaptive Learning: A key innovation of our system is the automated feedback loop. The system does not remain static; high-confidence predictions from the Data Model are used to update the abbreviation dictionary, thereby continuously improving the lightweight Metadata Model's performance over time.

5.2. Operational Impact

The deployment of this framework has yielded tangible operational benefits. The primary impact is the significant reduction in manual effort required for schema mapping. By generating automated scorecards and mapping sheets, the frequency of synchronization meetings and email exchanges between asset teams and CDL engineers has been drastically reduced. This shift allows domain experts to focus on validating difficult edge cases rather than performing rote mapping of obvious columns.

5.3. Future Work.

Future work includes developing a User Interface (UI) for self-service gap analysis, integrating with enterprise data governance tools, and establishing a "Golden Training Dataset" to improve the robustness of the KNN classifier.

References

[1] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," The VLDB Journal, vol. 10, no. 4, pp. 334–350, 2001.

[2] W. E. Winkler, "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage," Proceedings of the Section on Survey Research Methods, American Statistical Association, pp. 354–359, 1990.

[3] J. Madhavan, P. A. Bernstein, and E. Rahm, "Generic schema matching with Cupid," in Proceedings of the 27th International Conference on Very Large Data Bases (VLDB), 2001, pp. 49–58.

[4] A. Doan, P. Domingos, and A. Y. Halevy, "Reconciling schemas of disparate data sources: A machine-learning approach," in Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, 2001, pp. 509–520.

[5] Y. Zhang, J. He, and J. Liu, "Schema Matching using Machine Learning," in International Conference on Computer Science, 2011.

[6] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, 1967.

[7] H. Nottelmann and N. Fuhr, "Evaluating different methods of estimating retrieval quality for IIR," in Proceedings of the 26th Annual International ACM SIGIR Conference, 2003.

[8] M. Stonebraker, D. Bruckner, I. F. Ilyas, G. Beskales, M. Cherniack, S. B. Zdonik, A. Pagan, and S. Xu, "Data Curation at Scale: The Data Tamer System," in CIDR, 2013.

[9] R. Dhamdhere, H. Pethe, and A. K. N. N., "Managing Data Lakes at Scale," IEEE Data Engineering Bulletin, 2019.

[10] F. Panse, A. Griewank, and N. Ritter, "Schema-based Data Deduplication," IEEE Transactions on Knowledge and Data Engineering, 2020.