Intelligent Multi-Source Data Integration Framework Using Machine Learning for Schema Matching and Data Transformation

Pooja M V [1], Neelambike S [2], Karthik More P R [3]

[1] Assistant Professor, Department of Information Science and Engineering, GM University, Davanagere,

[2] Associate Professor and HOD Department of Information Science and Engineering, GM University, Davanagere,

[3] PG Student, Department of Data Engineering, Faculty of Engineering and Technology, GM School of Advanced Studies, GM University, Davanagere

---

Abstract

The increasing volume of data generated by heterogeneous information systems has created significant challenges in data integration, particularly when datasets contain different column names, data types, formats, and structural representations. Traditional data integration approaches rely heavily on manually defined mapping rules and predefined transformation procedures, making them difficult to maintain when integrating data from multiple sources. This paper proposes an Intelligent Multi-Source Data Integration Framework Using Machine Learning for Schema Matching and Data Transformation. The proposed framework aims to automate the identification of semantically equivalent attributes across multiple datasets by combining lexical, structural, statistical, and semantic features. A machine learning-based schema matching model estimates the similarity between source attributes and recommends corresponding target attributes using confidence scores. Based on the identified mappings, a transformation engine standardizes data formats, resolves selected inconsistencies, and combines compatible records into a unified dataset. The framework also incorporates data profiling, data quality assessment, transformation validation, and reporting mechanisms to improve integration reliability. The proposed system is designed to support structured data sources, including CSV files, Excel spreadsheets, and JSON datasets, with the potential for extension to relational databases. Its effectiveness can be evaluated using schema matching precision, recall, F1-score, transformation accuracy, execution time, and data quality improvement. The framework aims to reduce manual mapping effort, improve integration consistency, and provide a flexible foundation for intelligent data integration in data engineering applications.

Key Words: Multi-source data integration, machine learning, schema matching, data transformation, heterogeneous data, semantic similarity, data quality, ETL, automated data integration.

1. INTRODUCTION

Data integration is an essential process in modern data engineering that combines information from multiple independent sources into a consistent and meaningful representation. Organizations commonly store related information in different databases, spreadsheets, enterprise applications, and data exchange formats. Although these sources may describe the same real-world entities, their schemas and data representations frequently differ.

For example, a college may maintain student information in separate datasets containing attributes such as Student_ID, USN, StudentNumber, Student_Name, and FullName. Similarly, a banking system may store customer information using attributes such as Customer_ID, CIF_Number, and Client_Code. Integrating these datasets requires identifying equivalent attributes, standardizing their values, and resolving inconsistencies before generating a unified dataset.Traditional integration methods often depend on manually defined rules, exact column-name matching, and predefined transformation procedures. Such approaches can perform effectively when schemas remain stable, but they become less efficient when datasets contain unfamiliar column names, inconsistent data types, missing values, or different naming conventions. Machine learning offers an opportunity to improve this process by learning patterns that distinguish matching and non-matching attributes. By examining multiple characteristics of source columns, an intelligent system can recommend schema correspondences and support automated transformation decisions. This research proposes an intelligent framework that combines schema profiling, feature extraction, machine learning, transformation generation, and validation in a unified pipeline. The principal aim is to reduce manual intervention while maintaining transparency, reliability, and control over data integration operations.

1.1 Background /History

Data integration has evolved alongside the development of database systems and large-scale information processing.Initially, database integration relied on manually constructed queries, application-specific interfaces, and fixed mappings between data sources. As organizations adopted distributed databases and enterprise information systems, the need for common representations and interoperable schemas increased.Extract, Transform, and Load (ETL) processes subsequently became widely used in data warehousing. These processes extract information from source systems, transform it into a target representation, and load it into a destination repository. Although ETL supports repeatable integration workflows, the development and maintenance of transformation rules can require substantial manual effort.

Schema matching emerged as an important research area for identifying correspondences between database attributes and structures. Early approaches primarily used schema names, labels, data types, and structural relationships. Later approaches incorporated linguistic similarity, domain knowledge, instance-level statistics, and combinations of multiple matching techniques.With the growth of machine learning, schema matching increasingly became a problem of learning to distinguish corresponding attributes from non-corresponding ones. More recent semantic approaches use vector representations and language models to compare attributes whose names differ considerably but whose meanings are related.

The proposed framework builds on these developments by combining traditional schema-matching signals with machine

learning and rule-based transformation validation. Rather than treating matching and transformation as separate manual activities, it organizes them into a connected integration pipeline.

1.2 Existing Technology

Existing data integration technologies generally fall into the following categories.

1. **Rule-based schema matching:** Uses predefined rules, dictionaries, exact matches, and string similarity measures to identify equivalent attributes. It is simple to implement but may fail when unfamiliar synonyms or domain-specific terms are encountered.
2. **ETL-based integration:** Extracts data from multiple sources, applies transformations, and loads the results into a target system. Its effectiveness depends on the quality of the configured mappings and transformation logic.
3. **Statistical schema matching:** Compares attributes using data types, value distributions, uniqueness, null proportions, and other statistical properties. It can identify structural similarities but may confuse attributes that have similar distributions but different meanings.
4. **Semantic schema matching:** Uses semantic representations, ontologies, embeddings, or language models to identify attributes with related meanings. It can recognize more complex correspondences, although it may require additional computational resources and domain-specific validation.
5. **Machine-learning-based matching:** Learns from labelled attribute pairs or engineered features to predict whether two attributes correspond. Its performance depends on training data quality, feature selection, and the diversity of schemas encountered during evaluation.

These technologies provide valuable capabilities, but no single technique is guaranteed to handle every integration scenario. The proposed framework therefore combines multiple matching signals and validates generated transformations before producing the integrated output.

2. LITERATURE SURVEY

Schema matching has been studied extensively in database integration, data interoperability, and semantic information processing. Existing research explores linguistic matching, statistical techniques, neural networks, deep learning, and language-model-based approaches. The following studies provide a foundation for the proposed framework.

[1]. Rodrigues and da Silva – A study on machine learning techniques for the schema matching network problem (2021). The authors investigate schema matching across multiple related schemas rather than treating every comparison as an isolated pair. Their methods use machine learning and a bootstrapping strategy to reduce the dependence on manually labeled examples. The study is particularly relevant to multi-source integration because a network of schemas creates constraints that can improve matching quality. The reported results demonstrate that ML-based network matching can be competitive and can benefit from user feedback. This supports the proposed project's use of feedback and multi-source context.. Source

2. Zhang et al. (2023) — Schema Matching Using Pre-Trained Language Models. This study introduces Learned Schema Mapper (LSM), which uses pre-trained language models and active learning for schema matching. The approach addresses cases where attribute names differ semantically and where access to the underlying source data may be limited. Active learning helps reduce the amount of manual annotation required to identify useful schema correspondences.. Source

3. Wang, Li, and Wang (2022) — Sudowoodo: Contrastive Self-Supervised Learning for Multi-Purpose Data Integration and Preparation. Sudowoodo explores contrastive self-supervised learning for multiple data integration and preparation tasks. Its approach seeks to learn useful representations that can support tasks such as entity matching, data cleaning, and semantic type detection. The study demonstrates the relevance of shared representations for related data preparation problems. Source

4. Tu et al. (2023) — Unicorn: A Unified Multi-Tasking Model for Supporting Matching Tasks in Data Integration Unicorn investigates a unified model for supporting multiple matching tasks in data integration rather than requiring a separate model for each task. The research is relevant to systems that need to support different forms of matching across heterogeneous datasets. Source

5. Zhu et al. (2024) — A Survey of Knowledge Graph Entity Alignment Research Based on Graph Embedding. This survey reviews graph-embedding techniques for aligning entities across knowledge graphs. It examines approaches that represent entities and their relationships in a shared vector space. Although knowledge graph alignment differs from tabular schema matching, the work provides useful insights into using contextual relationships to improve matching decisions. Source



Table 1: Representative literature and design implications

3. PROBLEM STATEMENT

Organizations maintain related information in heterogeneous datasets whose schemas differ in naming conventions, data types, formats, and structural representations. Manual integration requires users to identify equivalent attributes, define mappings, standardize values, and validate the resulting data. This process is time-consuming and can introduce mapping errors, inconsistent transformations, and data quality problems.

Existing matching techniques may perform poorly when exact names are unavailable, semantic relationships are unclear, or source data contains substantial inconsistencies. Furthermore, identifying corresponding attributes does not automatically guarantee that their values can be safely transformed and combined.Therefore, the research problem is to develop an intelligent multi-source data integration framework that learns schema correspondences from multiple features, generates appropriate data transformations, and validates the integrated output while minimizing manual mapping effort.

The key problems addressed by this project are:

1. **Heterogeneous Data Sources:** Integrating datasets from different sources with varying formats, structures, and data types.
2. **Schema Mismatch:** Identifying equivalent attributes when column names differ, such as Student_ID, USN, and Student Number**.**
3. **Semantic Differences:** Recognizing attributes with similar meanings but different names, such as Birth_Date and Date Of Birth.
4. **Manual Mapping Effort:** Reducing the time and effort required to define schema mappings manually.
5. **Inconsistent Data Formats:** Standardizing date formats, data types, column names, and categorical values across datasets.
6. **Poor Data Quality:** Detecting missing values, duplicate records, inconsistent values, and invalid data during integration.
7. **Limited Adaptability of Traditional Methods:** Addressing the limitations of fixed rules and exact string matching when new schemas or unfamiliar attributes are introduced.
8. **Integration Reliability:** Validating schema matches and transformations to reduce incorrect mappings and improve the consistency of the unified dataset.

Therefore, The project aims to automate schema matching and data transformation across heterogeneous data sources using machine learning, while improving data quality and reducing manual intervention.

3.1 Scope of the Project

The project "Intelligent Multi-Source Data Integration Framework Using Machine Learning for Schema Matching and Data Transformation" focuses on developing an intelligent data integration system that can combine data from multiple heterogeneous sources into a unified and consistent format. The system uses Machine Learning (ML) techniques to automatically identify relationships between different schemas and support appropriate data transformation.

The scope of the project includes:

1. **Multi-Source Data Collection**: Integrate data from heterogeneous sources such as CSV files, Excel files, relational databases, JSON files, and APIs.
2. **Data Profiling and Analysis:** Analyze source datasets to identify attributes, data types, missing values, duplicates, inconsistencies, and structural differences.
3. **Automated Schema Matching:** Apply ML/NLP-based techniques to identify semantically similar attributes across different datasets, even when attribute names are different.
4. **Schema Mapping:** Generate mappings between source attributes and a target unified schema based on attribute names, data types, values, and semantic similarity.
5. **Data Transformation:** Transform data into a common structure by handling differences in formats, data types, units, naming conventions, and representations.
6. **Data Quality Improvement:** Detect and handle missing values, duplicate records, inconsistent formats, and invalid data during the integration process.
7. **Machine Learning-Based Matching:** Use suitable ML models such as Random Forest, XGBoost, similarity-based models, or transformer/NLP embeddings to improve schema matching accuracy.
8. **Unified Data Repositor:** Store the transformed and integrated data in a centralized database or data warehouse for further analysis.
9. **Integration Monitoring and Reporting:** Provide reports or dashboards showing schema matches, transformation activities, data-quality issues, and integration results.
10. **Scalability:** Design the framework so that additional datasets and data sources can be incorporated without redesigning the complete integration system.

3.2 Objectives

1. **To develop a multi-source data ingestion** module The objective is to develop a data ingestion module capable of importing heterogeneous structured datasets from multiple sources, such as CSV files, Excel spreadsheets, JSON files, and relational databases. The module will examine the structure of each dataset, identify column names, infer data types, and extract basic schema information. It will also perform initial checks to identify invalid files, missing columns, and incompatible data formats. This process establishes a standardized input for subsequent schema matching and transformation operations.
2. **To design a machine-learning-based schema matching model** The objective is to design a machine-learning model that automatically identifies equivalent attributes across datasets, even when their column names and representations differ. The model will use multiple features, including lexical similarity, structural information, statistical characteristics, and semantic similarity, to determine whether two attributes represent the same concept. For example, the attributes Student_ID, USN, and StudentNumber may represent the same student identifier. The model will generate matching predictions and confidence scores to support reliable schema alignment and identify uncertain mappings for further verification.
3. **To automate data transformation** The objective is to automate the transformation of matched attributes into a common target schema. The transformation module will standardize column names, convert compatible data types, normalize date formats, and correct selected inconsistencies using predefined transformation rules. For example, dates represented as 12/05/2025, 2025-05-12, and other supported formats can be converted into a common format after

their interpretation has been verified. The module will also maintain records of transformation operations and flag values that cannot be safely converted. This process reduces repetitive manual work and helps produce consistent integrated datasets.

4. **To improve data quality and integration reliability** The objective is to improve the quality and reliability of integrated data through systematic validation and data quality checks. The framework will identify missing values, duplicate records, invalid values, incompatible data types, and inconsistencies between source and target schemas. It will validate transformed values against predefined rules and report any errors detected during integration. Where appropriate, the system may apply approved data-cleaning procedures, while ambiguous or potentially destructive corrections are flagged for manual review. These measures help preserve data integrity and increase confidence in the resulting unified dataset.
5. **To evaluate the proposed framework** The objective is to evaluate the performance and effectiveness of the proposed framework using quantitative metrics and comparisons with existing techniques. Schema matching performance will be measured using precision, recall, and F1-score. Transformation accuracy will assess whether the generated transformations produce the expected results, while execution time will measure processing efficiency. Data quality measurements will be used to assess changes in missing values, duplicates, and consistency before and after integration. The proposed framework will also be compared with traditional rule-based and similarity-based matching approaches using the same evaluation datasets. The results will help identify the strengths, limitations, and potential improvements of the proposed system.

4. IMPLEMENTATION

Your implementation explains how the Intelligent Multi-Source Data Integration Framework Using Machine Learning for Schema Matching and Data Transformation is developed, from importing datasets to generating a validated integrated dataset..



Fig 4.1 Implementation Modules

The following describes the intended implementation; verification of the complete system and comparative experimental evaluation remain pending.

1. Implementation Modules

Module 1: Data Ingestion Module

Purpose: To collect and import data from different sources.

- Imports CSV, Excel, JSON, and database files.
- Validates file formats and structures.
- Extracts schema information such as column names, data types, and sample values.

Example: One dataset contains Student_ID, while another contains USN. The system imports both datasets for further processing.

Output: Imported datasets with their initial schema information.

Module 2: Schema Profiling Module

Purpose: To analyze the structure and quality of the imported datasets.

- Analyzes column statistics, including missing values, unique values, and value distributions.
- Identifies data types and patterns.
- Generates a schema summary for each dataset.

Example: The system identifies that Age contains numerical values, while Date Of Birth contains date values.

Output: A schema profile containing column properties and data quality information.

Module 3: Schema Matching Module

Purpose: To identify equivalent attributes across different datasets using machine learning.

It uses four types of features:

- Lexical: Compares column names and words.
- Structural: Examines data types and relationships between attributes.
- Statistical: Compares value distributions, uniqueness, and other statistics.
- Semantic: Examines whether different column names have similar meanings.

The model, such as Random Forest, predicts potential matches and generates confidence scores.

Example: It may identify Student_ID, USN, and Student Number as equivalent attributes when the available evidence supports the match.

Output: Predicted schema mappings with confidence scores.

Module 4: Transformation Module

Purpose: To convert matched attributes into a consistent format.

- Standardizes data types and date formats.
- Applies approved transformation rules.
- Normalizes column names.
- Handles selected inconsistent values.

Example: The columns FullName, Student_Name, and Name can be mapped to the standard column student_name. Compatible date values can also be converted into a common format.

Output: Transformed datasets following a common target schema.

Module 5: Validation and Quality Check Module

Purpose: To verify the correctness and consistency of the transformed data.

- Detects missing values.
- Identifies duplicate records.
- Validates transformed data.
- Checks schema consistency.
- Generates a data quality report.

Example: If a date cannot be converted into the required format, the system flags the invalid value instead of silently accepting it.

Output: Validated integrated data and a quality report.

2. Output

The implementation produces four main outputs:

1. Integrated dataset: Data combined into a unified schema.
2. Mapping logs: Records of the source-to-target attribute mappings.
3. Transformation records: Details of the transformations applied.
4. Quality report: Summary of missing values, duplicates, inconsistencies, and validation results.

5. SYSTEM ARCHITECTURE

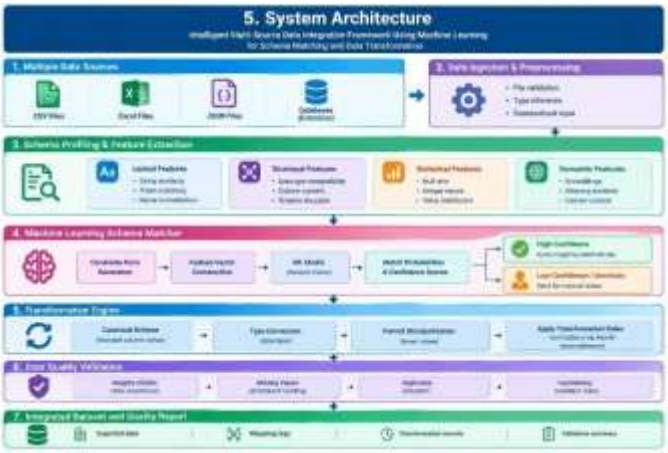


Fig 5.1 System Architecture

The system architecture of the Intelligent Multi-Source Data Integration Framework Using Machine Learning for Schema Matching and Data Transformation provides a systematic approach to integrating heterogeneous data from multiple sources into a unified and consistent dataset. Initially, the system collects structured data from CSV files, Excel spreadsheets, JSON files, and databases. The data ingestion and preprocessing module validates the input files, identifies data types, and prepares the datasets for further processing.

Next, the schema profiling and feature extraction module analyzes column names, data types, statistical properties, structural relationships, and semantic similarities to identify useful characteristics of the input datasets. These features are provided to a machine-learning-based schema matching model, such as Random Forest, which predicts corresponding attributes and generates confidence scores. High-confidence matches can be accepted automatically, while uncertain matches are flagged for manual review to reduce incorrect mappings.

After schema matching, the transformation engine converts the matched attributes into a common target schema by standardizing column names, converting compatible data types, normalizing date formats, and applying appropriate transformation rules. The data quality validation module then checks data integrity, missing values, duplicate records, and schema consistency to identify potential errors and inconsistencies. Finally, the framework generates an integrated dataset along with mapping logs, transformation records, and a quality report. Overall, the architecture aims to reduce manual effort, improve schema matching accuracy, maintain data consistency, and provide a reliable and traceable solution for multi-source data integration.

6. EXISTING MODEL

Traditional data integration models are widely used to combine data from multiple heterogeneous sources, such as CSV files, Excel spreadsheets, JSON files, and relational databases. These models generally rely on rule-based schema matching, similarity-based techniques, and predefined Extract, Transform, and Load (ETL) processes to identify corresponding attributes and integrate data into a common structure. In rule-based approaches, predefined conditions and mapping rules are used to match source attributes with target attributes. Similarity-based approaches compare column names, data types, and selected data characteristics to identify potential correspondences. Once the mappings are established, ETL processes perform data transformation, standardization, and loading into the target system.

Although these approaches provide essential data integration capabilities, they have several limitations. Rule-based methods require manual configuration and maintenance, particularly when new datasets introduce unfamiliar column names or different structural representations. Similarity-based methods may fail to identify semantically equivalent attributes when their names differ significantly. Furthermore, traditional ETL workflows often depend on predefined transformation rules, making them less adaptable to changing schemas and inconsistent data formats. Data quality problems, including missing values, duplicate records, incompatible data types, and inconsistent values, can also reduce the reliability of the integrated output. In addition, schema matching and data transformation are often handled as separate tasks, which can make end-to-end validation more challenging.

7. PROPOSED METHODOLOGY



Fig 7.1 Proposed Methodology

The proposed methodology for the "Intelligent Multi-Source Data Integration Framework Using Machine Learning for Schema Matching and Data Transformation" follows a systematic, step-by-step approach to integrate heterogeneous data sources into a unified and high-quality dataset.

Initially, data is collected from multiple sources, such as CSV, Excel, and JSON files. The collected datasets undergo data profiling to examine column names, data types, missing values, distinct values, and statistical patterns. Next, relevant lexical, structural, statistical, and semantic features are extracted to identify relationships between columns.

A Random Forest machine learning model is trained using labeled data to predict schema matches and assign confidence scores. High-confidence matches can be processed automatically, while uncertain matches are flagged for review. The matched columns are then transformed by standardizing column names, data types, date formats, and inconsistent categorical values.

After transformation, the datasets are integrated and validated to check schema consistency, missing values, duplicates, and transformation correctness. Finally, the framework evaluates its performance using precision, recall, F1-score, transformation accuracy, execution time, and data quality metrics. The final output includes a unified dataset, schema-mapping records, transformation logs, and a quality report

8. RESULT



Fig 8.1 Intelligent Multi-Source Data Integration Dashboard

The system integrates data from three different sources, such as CRM, shop, and other customer datasets. These sources may contain duplicate customer records, missing values, inconsistent formats, and different column names. The dashboard displays a total of 1,126 source records, which are consolidated into 568 golden customer records after merging duplicate records.



Fig 8.2 Schema Matching

The Schema Matching module is a core component of the proposed framework. Its primary purpose is to identify relationships between columns from different data sources and map them to a common or canonical customer schema. Since different systems may use different names for the same information, direct integration can be difficult without schema matching.



Fig 8.3 Review Queue

The framework uses two configurable confidence thresholds: an auto-apply threshold of 0.70 and a review threshold of 0.25, as shown in the screenshot. Mappings with sufficiently high confidence can be applied automatically, while mappings falling into the review range are presented to a reviewer. Mappings below the review threshold can be treated as unreliable or left unmapped, depending on the implementation.

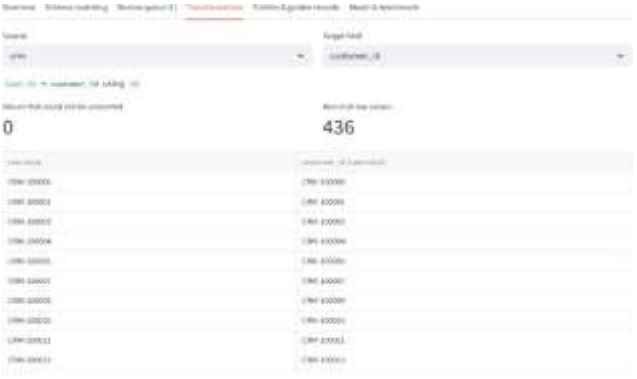


Fig 8.4 Transformations

The Data Transformation module maps source fields to standardized target fields. In the given image, the CRM field Cust_ID is mapped to customer_id using identity transformation. The system processes 436 non-null values with zero conversion errors, demonstrating successful field mapping and data standardization..

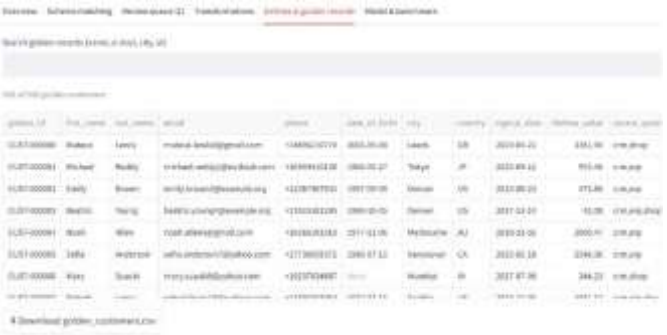


Fig 8.4 Entities & Golden Records

The Entities and Golden Records module integrates customer information from multiple sources, such as CRM, ERP, and shopping systems, into a unified dataset. It contains 568 golden customer records with details such as names, email addresses, phone numbers, and locations. The integrated data can be exported as a CSV file for further analysis.

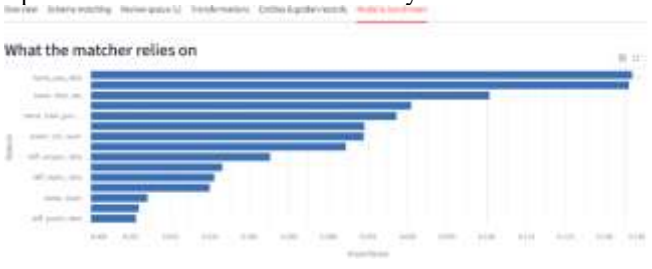


Fig 8.4 Final Model & Benchmark

The Model and Benchmark module displays the importance of different features used by the machine learning model for schema matching. Features such as sequence similarity, character similarity, token coverage, and data-value characteristics help identify matching columns across datasets. The chart indicates that name-based similarity features have a significant influence on the model's matching decisions.

9. CONCLUSIONS

This paper presents the design of an Intelligent Multi-Source Data Integration Framework Using Machine Learning for Schema Matching and Data Transformation. The proposed framework addresses the challenges of integrating heterogeneous structured datasets with inconsistent attribute names, data types, formats, and representations.

By combining schema profiling, lexical and statistical analysis, structural information, and optional semantic features, the framework aims to identify corresponding attributes more intelligently than approaches based exclusively on predefined rules. The integration pipeline connects matching predictions with controlled data transformation, quality validation, and reporting.The use of confidence-based decisions and human review provides a mechanism for handling ambiguous mappings, while transformation logs support traceability and reproducibility. The proposed implementation can be developed using Python, Pandas, scikit-learn, Streamlit, and SQLite. Future work may explore transformer-based semantic matching, active learning from user corrections, incremental integration for evolving schemas, and large-scale processing of additional data formats. A systematic comparison with established matching techniques is necessary to quantify the framework's benefits, limitations, and generalization capability.

.

ACKNOWLEDGEMENT

The author expresses sincere gratitude to the project guide, Mrs. Niveditha N M, for guidance and constructive feedback during the development of this work. The author also thanks the faculty and staff of GM University, Davangere, for their academic support. The contributions of the researchers and open-source developers whose work informed the proposed self-healing data pipeline are gratefully acknowledged.

REFERENCES

A. Journal Articles

[1] D. Rodrigues and A. da Silva, “A study on machine learning techniques for the schema matching network problem,” Journal of the Brazilian Computer Society, vol. 27, article 14, 2021. DOI: 10.1186/s13173-021-00119-5. [2] C.-C. Sun and D.-R. Shen, “Mixed Hierarchical Networks for Deep Entity Matching,” Journal of Computer Science and Technology, vol. 36, pp. 822–838, 2021. DOI: 10.1007/s11390-021-1321-0.

[3] E. Krivosheev, M. Atzeni, K. Mirylenka, P. Scotton, C. Miksovic, and A. Zorin, “Business Entity Matching with Siamese Graph Convolutional Networks,” AAAI Conference on Artificial Intelligence, 2021.

[4] A. Bogatu, N. W. Paton, M. Douthwaite, S. Davie, and A. Freitas, “Cost-effective Variational Active Entity Resolution,” IEEE International Conference on Data Engineering (ICDE), 2021. DOI: 10.1109/ICDE51399.2021.00114.

[5] A. Jain, S. Sarawagi, and P. Sen, “Deep Indexed Active Learning for Matching Heterogeneous Entity Representations,” Proceedings of the VLDB Endowment, vol. 15, no. 1, pp. 31–45, 2021. DOI: 10.14778/3485450.3485455.

[6] B. Li, Y. Miao, Y. Wang, Y. Sun, and W. Wang, “Improving the Efficiency and Effectiveness for BERT-based Entity Resolution,” Proceedings of AAAI, vol. 35, no. 15, pp. 13226–13233, 2021. DOI: 10.1609/aaai.v35i15.17562.

[7] P. Wang, W. Zheng, J. Wang, and J. Pei, “Automating Entity Matching Model Development,” IEEE ICDE, 2021. DOI: 10.1109/ICDE51399.2021.00116.

[8] W. Ren, X. Lian, and K. Ghazinour, “Online Topic-Aware Entity Resolution Over Incomplete Data Streams,” SIGMOD 2021, pp. 1478–1490. DOI: 10.1145/3448016.3457238.

[9] N. Barlaug and J. A. Gulla, “Neural Networks for Entity Matching: A Survey,” ACM Transactions on Knowledge Discovery from Data, vol. 15, no. 3, 2021. DOI: 10.1145/3442200.

[10] R. Peeters and C. Bizer, “Supervised Contrastive Learning for Product Matching,” Companion Proceedings of the Web Conference 2022, pp. 248–251. DOI: 10.1145/3487553.3524254.

[11] C. Ye, S. Jiang, H. Zhang, Y. Wu, J. Shi, H. Wang, and G. Dai, “JointMatcher: Numerically-aware entity matching using pre-trained language models with attention concentration,” Knowledge-Based Systems, vol. 251, article 109033, 2022.

[12] M. Trabelsi, J. Heflin, and J. Cao, “DAME: Domain Adaptation for Matching Entities,” WSDM 2022. DOI: 10.1145/3488560.3498486.

[13] J. Tu et al., “Domain Adaptation for Deep Entity Resolution,” Proceedings of the ACM SIGMOD International Conference 10.1145/3514221.3517870. on Management of Data, pp. 443–457, 2022. DOI:

[14] D. Yao, Y. Gu, G. Cong, H. Jin, and X. Lv, “Entity Resolution with Hierarchical Graph Attention Networks,” SIGMOD 2022. DOI: 10.1145/3514221.3517872.

[15] S. Galhotra, D. Firmani, B. Saha, and D. Srivastava, “Hierarchical Entity Resolution using an Oracle,” SIGMOD 2022, pp. 414–428. DOI: 10.1145/3514221.3526147.

[16] J. Wang, Y. Li, W. Hirota, and E. Kandogan, “Machop: an End-to-End Generalized Entity Matching Framework,” aiDM 2022. DOI: 10.1145/3533702.3534910.

[17] R. Wang, Y. Li, and J. Wang, “Sudowoodo: Contrastive Self-supervised Learning for Multi purpose Data Integration and Preparation,” arXiv:2207.04122, 2022.

[18] Y. Zhang, A. Floratou, J. Cahoon, S. Krishnan, A. C. Müller, D. Banda, F. Psallidas, and J. M. Patel, “Schema Matching using Pre-Trained Language Models,” IEEE ICDE, 2023.

[19] Y. Li, J. Li, Y. Suhara, A. Doan, and others, “Effective Entity Matching with Transformers,” The VLDB Journal, vol. 32, 2023. DOI: 10.1007/s00778-023-00779-z.

[20] J. Huang, Z. Sun, Q. Chen, X. Xu, W. Ren, and W. Hu, “Deep Active Alignment of Knowledge Graph Entities and Schemata,” Proceedings of the ACM on Management of Data, vol. 1, no. 2, article 159, 2023. DOI: 10.1145/3589304.

[21] J. Tu et al., “Unicorn: A Unified Multi-tasking Model for Supporting Matching Tasks in Data Integration,” Proceedings of the ACM on Management of Data, vol. 1, no. 1, 2023. DOI: 10.1145/3588938.

[22] B. Genossar, R. Shraga, and A. Gal, “FlexER: Flexible Entity Resolution for Multiple Intents,” Proceedings of the

ACM on Management of Data, vol. 1, no. 1, article 42, 2023. DOI: 10.1145/3588722.
[23] R. Wu, A. E. Bendeck, X. Chu, and Y. He, "Ground Truth Inference for Weakly Supervised Entity Matching," SIGMOD 2023.
[24] Z. Pan, M. Yang, and A. Monti, "Schema Matching based on Energy Domain Pre-trained Language Model," Energy Informatics, vol. 6, article 22, 2023. DOI: 10.1186/s42162-023-00277-0.
[25] M. Masmoudi, S. B. A. B. Lamine, M.-H. Karray, B. Archimède, and H. B. Zghal, "Semantic Data Integration and Querying: A Survey and Challenges," ACM Computing Surveys, vol. 56, no. 8, article 209, 2024. DOI: 10.1145/3653317.
[26] B. Zhu et al., "A survey: knowledge graph entity alignment research based on graph embedding," Artificial Intelligence Review, vol. 57, article 229, 2024. DOI: 10.1007/s10462-024-10866-4.
[27] Q. Huang and T. Zhao, "Leveraging Large Language Models for Entity Matching," arXiv:2405.20624, 2024. [28] Y. Xu, H. Li, K. Chen, and L. Shou, "KcMF: A Knowledge-compliant Framework for Schema and Entity Matching with Fine-tuning-free LLMs," arXiv:2410.12480, 2024.
[29] J. F. Low, B. C. M. Fung, and P. Xiong, "Better entity matching with transformers through ensembles," Knowledge-Based Systems, vol. 293, article 111678, 2024. DOI: 10.1016/j.knosys.2024.111678.
[30] H. Ding, C. Dai, Y. Wu, W. Ma, and H. Zhou, "SETEM: Self-ensemble training with Pre-trained Language Models for Entity Matching," Knowledge-Based Systems, vol. 293, article 111708, 2024. DOI: 10.1016/j.knosys.2024.111708.
[31] M. Raoui, M. Ennaouri, M. H. el Yazidi, and A. Zellou, "SchemaLogix: Advancing Interoperability with Machine Learning in Schema Matching," International Journal of Advanced Computer Science and Applications, vol. 15, no. 5, 2024. DOI: 10.14569/IJACSA.2024.0150542.
[32] "Leveraging Machine Learning For Enhanced Database Integration," Procedia Computer Science, vol. 235, pp. 1623–1633, 2024. DOI: 10.1016/j.procs.2024.04.154.
[33] N. Seedat and M. Van der Schaar, "Bootstrapping Self-Improvement of Language Model Programs for Zero-Shot Schema Matching," Proceedings of the 42nd International Conference on Machine Learning, PMLR 267, pp. 53791–53826, 2025.