Data Quality Management Tools
A Comparative Literature Review

Kristóf Balázs, Stefanos Kypritidis

Erasmus Mundus Joint Master Degree Programme in Big Data Management and Analytics (BDMA)

eBISS 2025

Introduction

High-quality data is important for reliable analysis in big data and data science. However, ensuring good data quality is challenging due to its multidimensional nature:

- **Dimensions:** *Accuracy*, *completeness*, *consistency*, *timeliness*, and *accessibility*. Each of these can vary depending on context and application.
- **Common Errors:** Missing values, duplicates, outliers, inconsistencies, and violations of data integrity constraints.
- **Challenges:** Data continuously changing, heterogeneous sources, no standardized metrics.

Different automated error-detection approaches are available, with each having strengths and limitations:

- **Rule-based Systems:** Perform well in domain specific contexts ***but*** have limited generalization.
- **Statistical & ML Approaches:** Generalize well across datasets ***but*** usually have difficulties with interpretability and adaptability.
- **Knowledge-based Tools:** use external semantic resources well ***but*** they depend heavily on the quality of the underlying knowledge bases.

Our Contribution:

In this study, we **compare leading data quality management tools based on their error-detection strategies**. Our aim is to highlight their strengths, limitations, and areas where different approaches complement each other. Ultimately, our goal is to help practitioners choose and combine tools in a way that best addresses the specific data quality challenges they face.

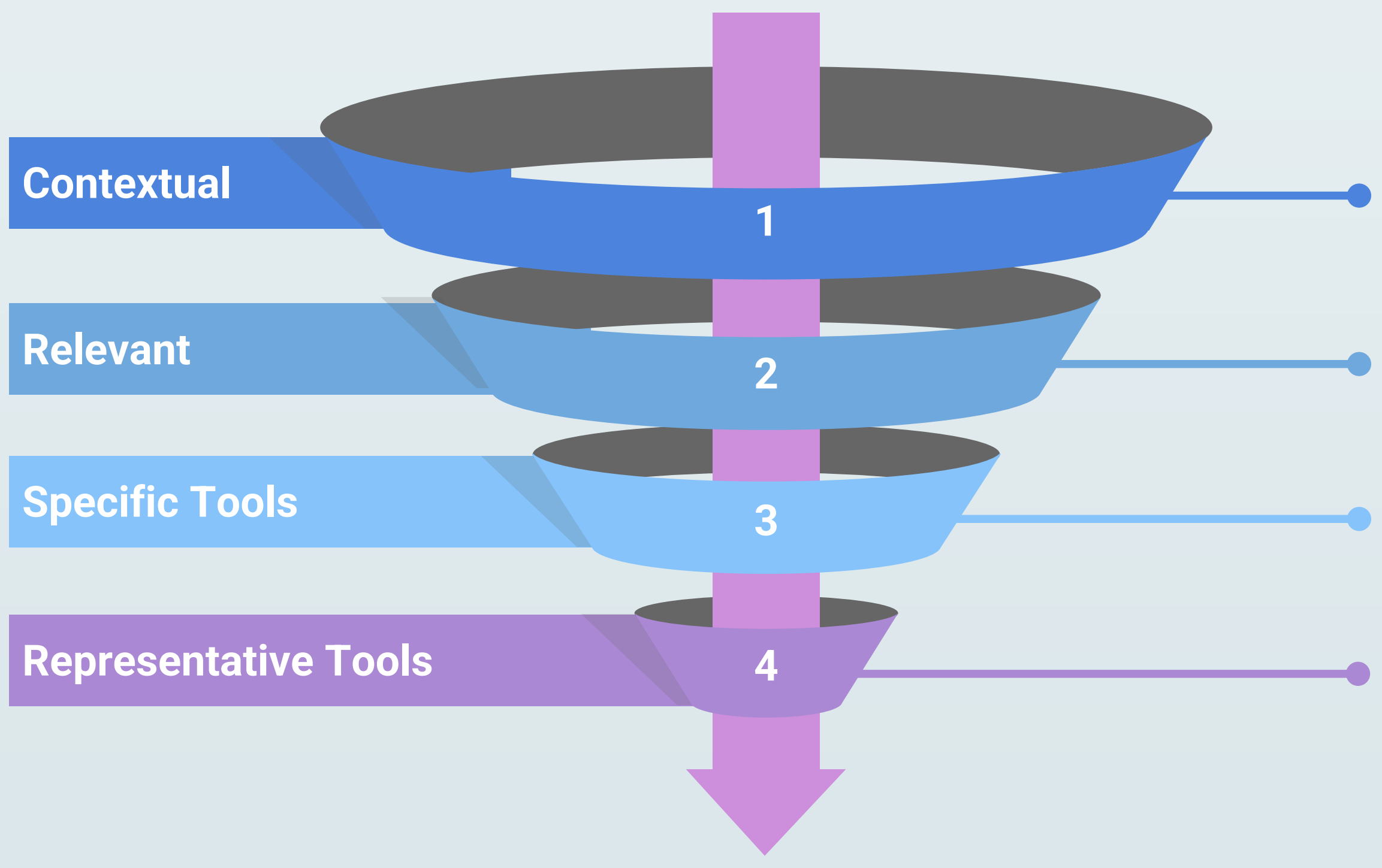
Methodology

This work uses a **comparative literature review** methodology, specifically **focusing on automated error detection tools and their strategies** within data quality (DQ) management.

The research question:

Which automated error-detection strategies do current data quality tools use, and how well does each strategy cover the main data quality dimensions and error types found in data?

Given the amount of research on data quality, **the scope was narrowed down the following way:**



General papers on data quality management (keyword-based search, snowballing, and filtering)

Litriture reviews, **surveys**, other papers on error detection methods, quality dimensions and tools

Papers introducing the **data quality tools** and error detection **methods**

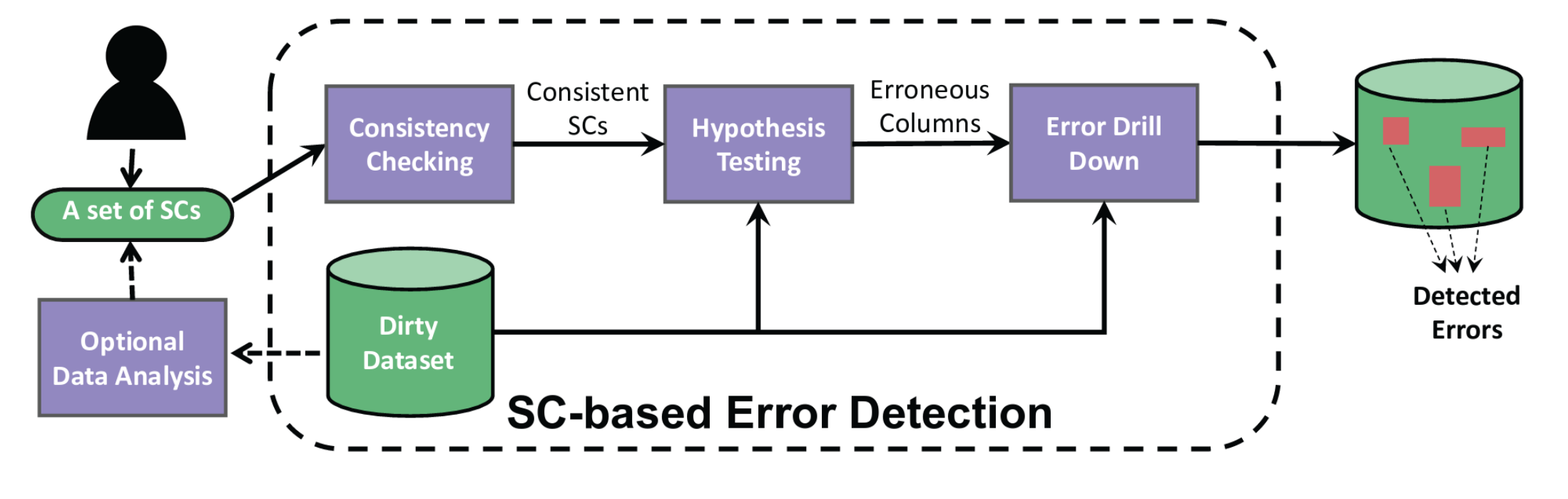
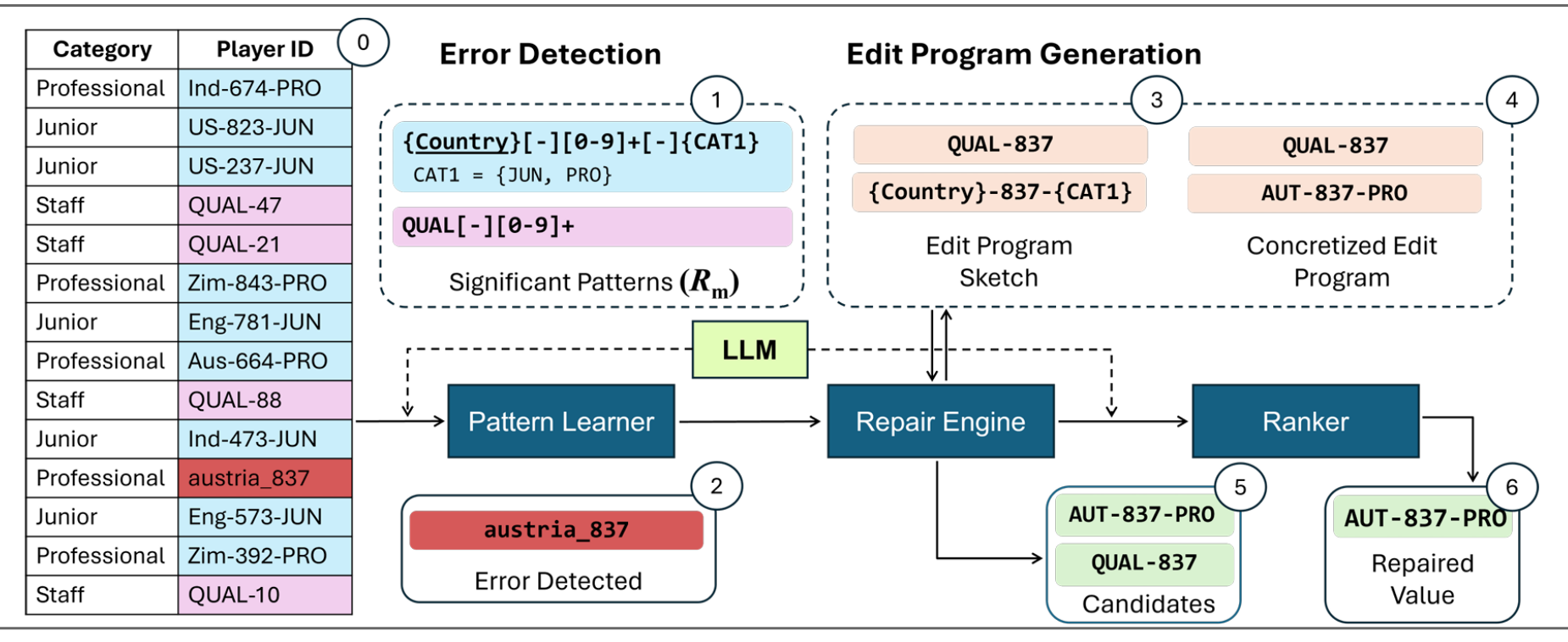
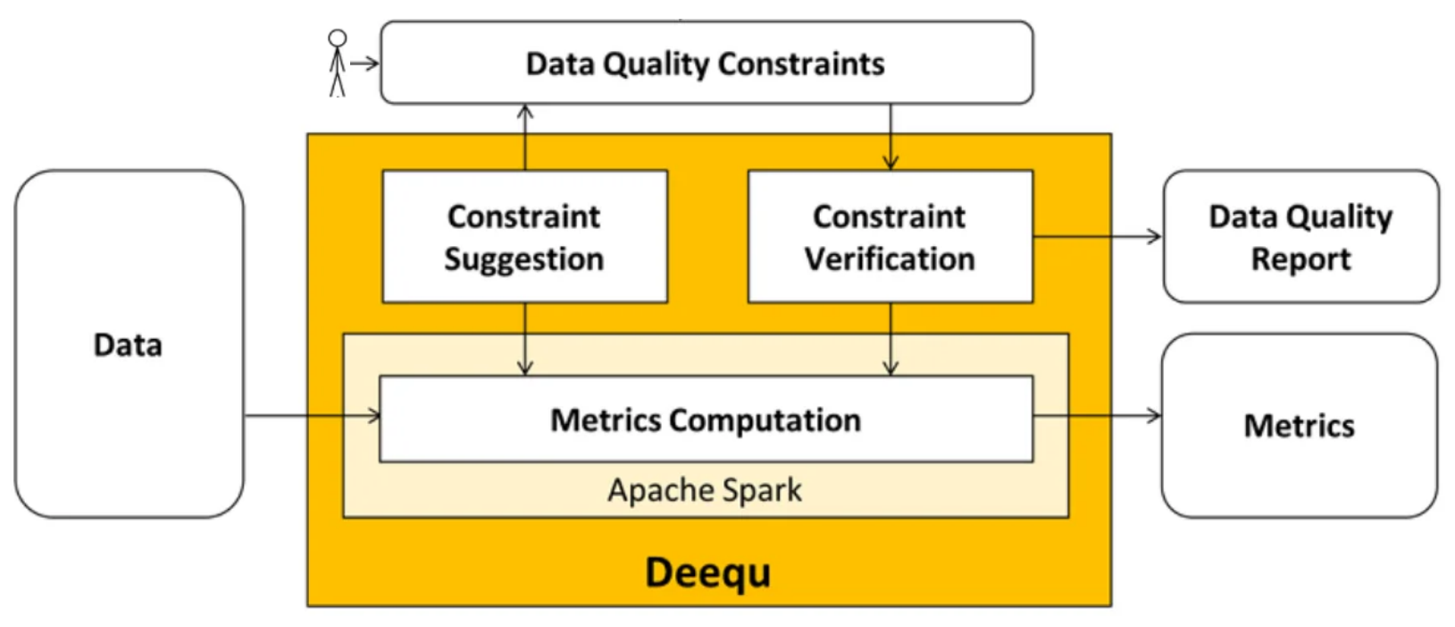
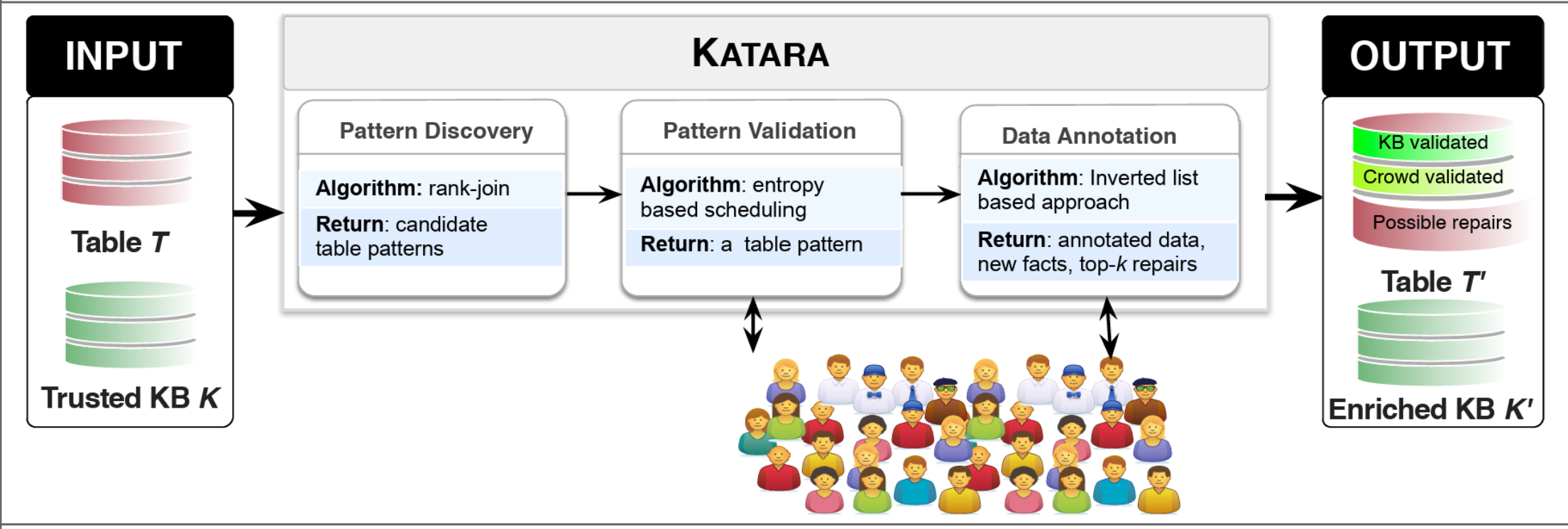
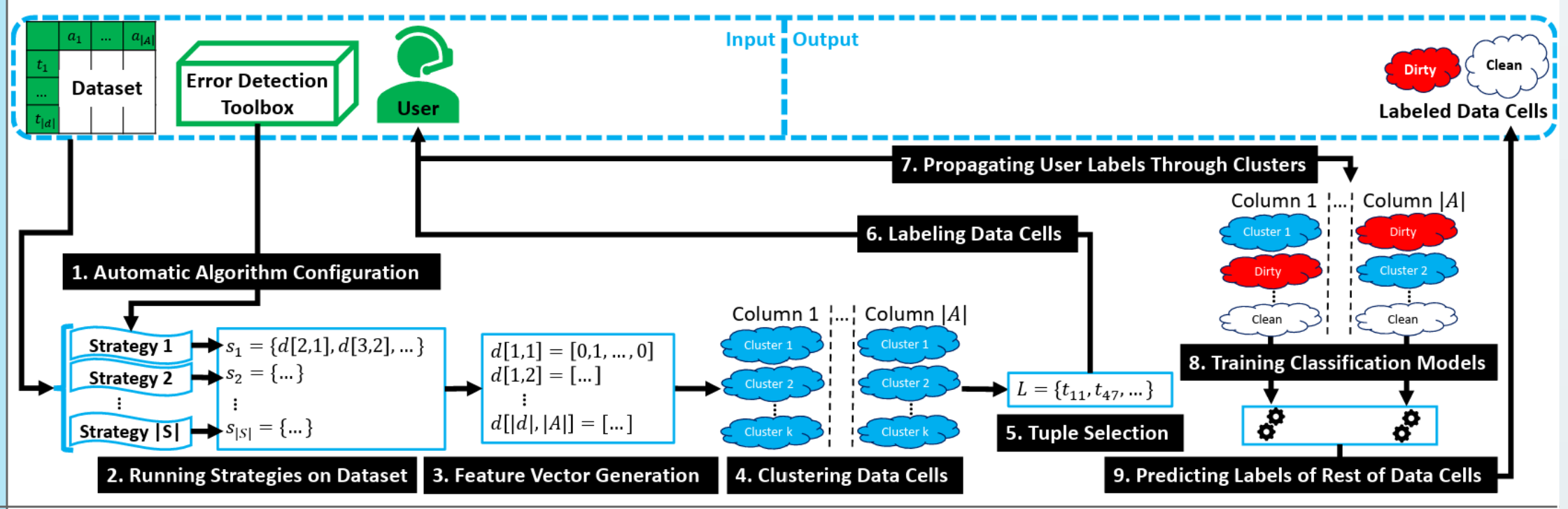
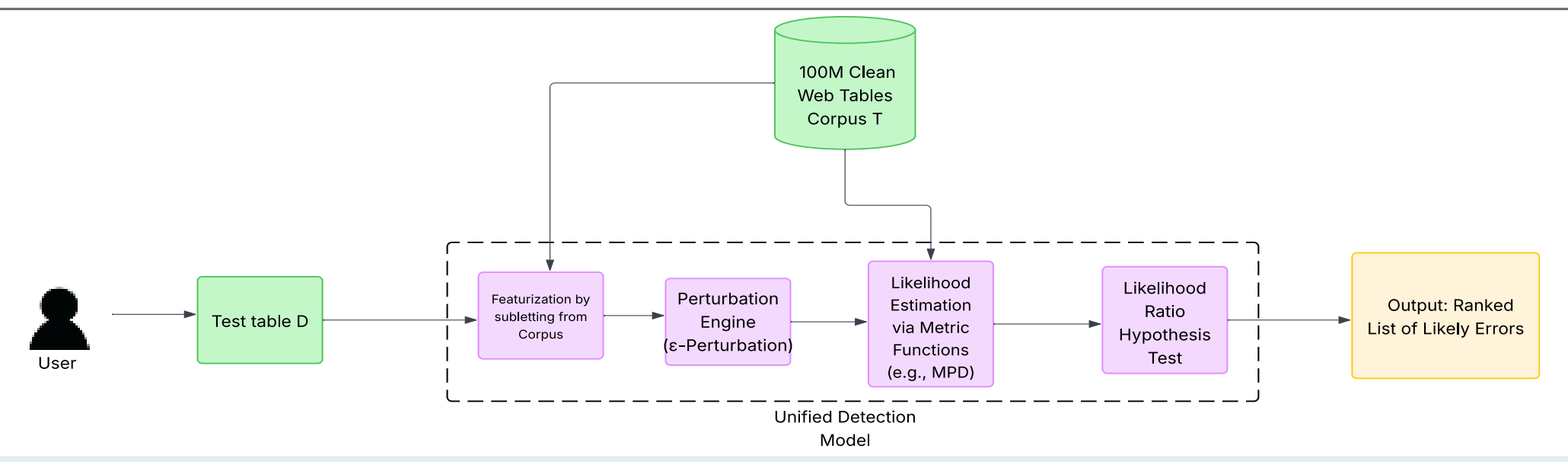
Constraint-based, Statistical-based, ML-based, Knowledge-based, Transformer-based

Results

Tool **comparison** matrix: 🟢 High, 🟡 Medium, 🔴 Low

Tool	Key Contribution	Access	Accuracy	Completeness	Consistency	Timeliness
CODED [1]	Statistical constraints for scalable error detection	🟡	🟢	🟡 (Outliers, missing values)	🟢 (statistical constraints)	🟡 (updating constraints)
DataVinci [2]	Unsupervised string error detection and repair with regex mining plus LLM reasoning	🟢	🟢 (str)	🔴 (focus on correcting strings)	🟡 (consistent patterns in strings)	🟡 (learning phase required)
Deequ [3]	Spark library; declarative DQ tests and automatic rule suggestion	🟡	🟢	🟡 (requires defined checks)	🟢 (intra & inter-column consistency)	🟢 (spark jobs / scheduled runs)
KATARA [4]	Combines KB alignment with crowd validation for semantic error repair	🟡	🟢	🟡 (limited scope)	🟡 (depends KB quality)	🟡 (crowdsource delays)
Raha [5]	Config-free detector that fuses 100+ base signals with minimal labels	🟢	🟢	🟡 (focus on correctness)	🟡 (bin. vector & classifier)	🟡 (offline processing)
Uni-Detect [6]	Unsupervised, schema-agnostic error detection without configuration	🟢	🟢	🟡 (focus on unseen errors)	🟡 (trained on clean tables)	🟡 (can not configure)

Methods & Architectures

Tool	Year	Method	Architecture
CODED	2019	Statistical & constraint	 A set of SCs Optional Data Analysis Consistency Checking Consistent SCs Hypothesis Testing Erroneous Columns Error Drill Down Dirty Dataset SC-based Error Detection Detected Errors
DataVinci	2025	Transformer-based pattern learning	 Category Player ID Error Detection Edit Program Generation {Country}[-][0-9]+[-]{CAT1} CAT1 = {JUN, PRO} QUAL[-][0-9]+ Significant Patterns (R_m) QUAL-837 {Country}-837-{CAT1} Edit Program Sketch AUT-837-PRO Concretized Edit Program LLM Pattern Learner Repair Engine Ranker austria_837 Error Detected Candidates Repaired Value
Deequ	2018	Constraint checks & profiling	 Data Quality Constraints Data Constraint Suggestion Constraint Verification Data Quality Report Metrics Computation Apache Spark Metrics Deequ
KATARA	2015	Knowledge-base & crowdsourcing	 INPUT Table T Trusted KB K KATARA Pattern Discovery Algorithm: rank-join Return: candidate table patterns Pattern Validation Algorithm: entropy based scheduling Return: a table pattern Data Annotation Algorithm: Inverted list based approach Return: annotated data, new facts, top-k repairs OUTPUT KB validated Crowd validated Possible repairs Table T' Enriched KB K'
Raha	2019	ML ensemble (active learning)	 Dataset Error Detection Toolbox User Input Output Dirty Clean Labeled Data Cells 1. Automatic Algorithm Configuration 2. Running Strategies on Dataset 3. Feature Vector Generation 4. Clustering Data Cells 5. Tuple Selection 6. Labeling Data Cells 7. Propagating User Labels Through Clusters 8. Training Classification Models 9. Predicting Labels of Rest of Data Cells Strategy 1 Strategy 2 Strategy |S| Column 1 Column |A| Cluster 1 Cluster 2 Cluster k
Uni-Detect	2019	Statistical & ML	 100M Clean Web Tables Corpus T User Test table D Featurization by subletting from Corpus Perturbation Engine (ε-Perturbation) Likelihood Estimation via Metric Functions (e.g., MPD) Likelihood Ratio Hypothesis Test Output: Ranked List of Likely Errors Unified Detection Model

Conclusions

Choosing the right data quality tool depends on multiple factors. Key considerations:

- No single tool can automatically detect and explain **all data quality** error types
- Each tool has **strengths** and **limitations**
- Most tools underperform in **timeliness** (except for Deequ)
- Tool selection **depends** on:

Domain of data

Data **structure**

Error types

Availability of **domain experts**

Operational constraints (scalability, computational resources, real-time needs, interpretability, privacy)

Future research should focus:

- **Real-Time** Error Detection which is critical for IoT and log monitoring
- Standardized **Benchmarks** & **Frameworks** to compare performance across different data quality dimensions
- Machine & Deep Learning approaches **interpretability** enhancements with interactive interfaces to increase **trustworthiness**

Data quality requires *tool diversity*; choose a tool that matches your use case and data.

References

[1] Yan et al. Detecting data errors with statistical constraints. arXiv:1902.09711, 2019.

[2] Singh et al. DataVinci: Learning syntactic and semantic string repairs. ACM Trans. Manag. Data, 3(1):1–26, 2025.

[3] Schmidt et al. Test data quality at scale with Deequ. AWS Blog, May 2019. [Accessed: 2025-06-16]

[4] Chu et al. Katara: A data cleaning system powered by knowledge bases and crowdsourcing. SIGMOD, 2015, pp. 1247–1261.

[5] Mahdavi et al. Raha: A configuration-free error detection system. SIGMOD, 2019, pp. 865–882

[6] Wang & He. Uni-Detect: A unified approach to automated error detection in tables. SIGMOD, 2019, pp. 811–828.