AutoDCWorkflow: LLM-based Data Cleaning Workflow Auto-Generation and Benchmark

Lan Li* **Liri Fang*** **Bertram Ludäscher** **Vetle I. Torvik**
School of Information Sciences
University of Illinois Urbana-Champaign
{lanl2, lirif2, ludaesch, vtorvik}@illinois.edu

Abstract

Data cleaning is a time-consuming and error-prone manual process, even with modern workflow tools like OpenRefine. Here, we present AutoDCWorkflow, an LLM-based pipeline for automatically generating data-cleaning workflows. The pipeline takes a raw table coupled with a data analysis purpose, and generates a sequence of OpenRefine operations designed to produce a minimal, clean table sufficient to address the purpose. Six operations address common data quality issues, including format inconsistencies, type errors, and duplicates.

To evaluate AutoDCWorkflow, we create a benchmark with metrics assessing answers, data, and workflow quality for 142 purposes using 96 tables across six topics. The evaluation covers three key dimensions: (1) **Purpose Answer**: can the cleaned table produce a correct answer? (2) **Column (Value)**: how closely does it match the ground truth table? (3) **Workflow (Operations)**: to what extent does the generated workflow resemble the human-curated ground truth? Experiments show that Llama 3.1, Mistral, and Gemma 2 significantly enhance data quality, outperforming the baseline across all metrics. Gemma 2-27B consistently generates high-quality tables and answers, while Gemma 2-9B excels in producing workflows that resemble human annotations.

1 Introduction

Data curation and cleaning are critical processes that prepare raw data for analysis, ensuring high quality and reliability (Chen et al., 2023). Implementing a sequence of data operations in the form of a data cleaning workflow improves data quality, enabling downstream analyses and yielding trustworthy results and actionable insights (Li et al., 2021; Wilkinson et al., 2016; Parulian and Ludäscher, 2022; McPhillips et al., 2019; Fang et al., 2025b). To facilitate automation and reuse,

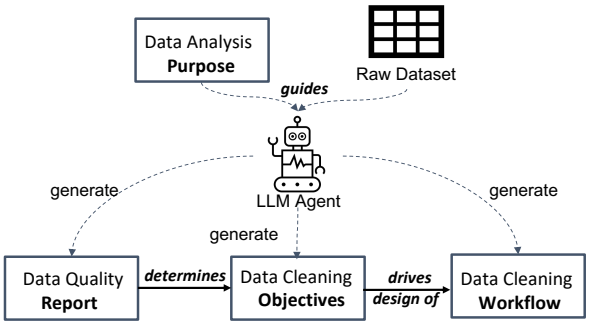

Figure 1: In a purpose-driven data cleaning model, AutoDCWorkflow leverages LLM agents to decide each reasoning step and automatically generate a data cleaning workflow based on a curator-defined analysis purpose and a provided dirty dataset.

tools like OpenRefine (Guidry, 2020) and Trifacta (Kandel et al., 2011) automatically capture reusable, executable workflows that ensure consistent results when re-applied to the same raw input.

Data cleaning should ensure data is fit-for-purpose, with the analysis purpose guiding the data quality report to identify relevant data quality issues (Wang and Strong, 1996; Pipino et al., 2002). These issues determine data cleaning objectives, which define actionable constraints and operations necessary to improve data quality. Consequently, data cleaning objectives drive the design of data cleaning workflows, as illustrated in Figure 1. Despite advancements, data scientists still spend over 80% of their time on cleaning tasks due to diverse domain requirements (Rezig et al., 2019). Designing effective workflows often involves multiple rounds of selecting appropriate operations and arguments to ensure accuracy (Stonebraker et al., 2018). This process remains time-consuming and error-prone, as it requires domain expertise and a deep understanding of the dataset and its schema, yet it is essential for auditing data analyses and ensuring the quality of the resulting tables.

Recent advances in large language models (LLMs) have sparked interest in their application to

*Equal contribution

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 7766–7780
November 4-9, 2025

Frameworks	Task	Workflow	Multi-DQ	Purpose
NADEEF (Dallachiesa et al., 2013)	Rule-based DC	✓	✓	
ActiveClean (Krishnan et al., 2016)	DC for ML		✓	
Holoclean (Rekatsinas et al., 2017)	ML for DC		✓	
SAGA (Siddiqi et al., 2023)	DC for ML	✓	✓	✓*
RetClean (Eltabakh et al., 2024)	LLM for DC		✓	
LLMClean (Biester et al., 2024)	LLM for DC		✓	
CleanAgent (Qi and Wang, 2024)	LLM for DC			✓
COMET (Mohammed et al., 2025)	DC for ML	✓	✓	
AutoDCWorkflow	LLM for DC	✓	✓	✓

Table 1: Comparison of existing data cleaning (DC) frameworks and **AutoDCWorkflow** in terms of whether the framework (1) generates an executable data cleaning workflow (**Workflow**), (2) supports multiple data quality issues (**Multi-DQ**), and (3) is guided by a specific data analysis purpose (**Purpose**). $*$ refers to the purpose being defined by downstream machine learning performance, rather than a data analysis purpose.

data cleaning (Eltabakh et al., 2024; Biester et al., 2024; Qi and Wang, 2024). However, few existing approaches effectively automate the design of executable data cleaning workflows that are aligned with specific analytical purposes and capable of addressing diverse data quality issues, despite the availability of modern cleaning tools (see Table 1). To address this gap, we propose AutoDCWorkflow pipeline and dataset that jointly fulfill three critical criteria: (1) is guided by a data analysis purpose, (2) supports the resolution of multiple data quality issues, and 3) generates executable and reusable data cleaning workflows.

Two natural questions arise in the context of LLM-assisted data cleaning: (1) Can LLMs help users design efficient, purpose-driven data cleaning workflows? (2) How can we evaluate LLMs' ability to automate these workflows? Therefore, we propose (1) an LLM-based pipeline, AutoDCWorkflow, that **Auto**matically generates a **D**ata **C**leaning **Workflow** given a data analysis purpose, and (2) a new dataset benchmark for evaluating automated workflow generation.

AutoDCWorkflow pipeline leverages LLMs as data cleaning agents to assess data quality and generate a sequence of operations based on the analysis purpose and raw table, inspired by the purpose-driven data cleaning model (Li and Ludäscher, 2023). It features three iterative, prompt-based components, as shown in Figure 2, and continues iterating until no data quality issues remain. Notably, AutoDCWorkflow is a companion tool built on OpenRefine. While the pipeline itself is programming language-agnostic, we use OpenRefine operations for demonstration in our experiments.

The benchmark evaluates LLM agents' ability to automate data cleaning workflows. AutoDCWorkflow provides annotated datasets that include purposes, answers, raw tables, manually curated clean tables, and workflows. The raw tables are sampled from six real-world datasets in domains like social science, public health, finance, and transportation.

Summary of Contributions. AutoDCWorkflow introduces a fully automated LLM-driven framework for data cleaning workflow generation. Our benchmark and evaluation metrics establish a foundation for future advancements in LLM-powered data cleaning. Our contributions include:

(1). We design a three-stage LLM-based pipeline—Select Target Columns, Inspect Column Quality, and Generate Operations & Arguments—for automated data cleaning workflow generation.

(2). We construct a dataset benchmark to evaluate data cleaning workflows across three dimensions: Purpose Answer, Column Value, and Workflow (Operations).

(3). We assess the performance of state-of-the-art models including Llama 3.1–8B, Mistral–7B, Gemma 2–9B and 27B. Gemma 2–9B and 27B stand out, consistently generating plausible workflows that closely resemble human-curated ones, resulting in more accurate purpose-driven answers.

2 Related Work

Data Cleaning for Machine Learning. Several systems have been developed to improve the quality of training data used in machine learning (ML) pipelines. NADEEF (Dallachiesa et al., 2013) provides a rule-based framework for detecting and repairing data errors using user-defined constraints. ActiveClean (Krishnan et al., 2016) integrates data cleaning with model training by prioritizing cleaning based on the model's sensitivity to dirty data. SAGA (Siddiqi et al., 2023) and COMET (Mohammed et al., 2025) extend this paradigm by generating executable workflows that address multiple types of data quality issues, optimizing cleaning strategies to maximize downstream model performance. In particular, SAGA advances fit-for-purposes data cleaning by systematically learning and optimizing cleaning strategies based on downstream ML performance. However, these frameworks are primarily designed to enhance model accuracy, rather than being tailored to specific user-defined analytical purposes.

Machine Learning for Data Cleaning. Machine learning techniques have long been used to sup-

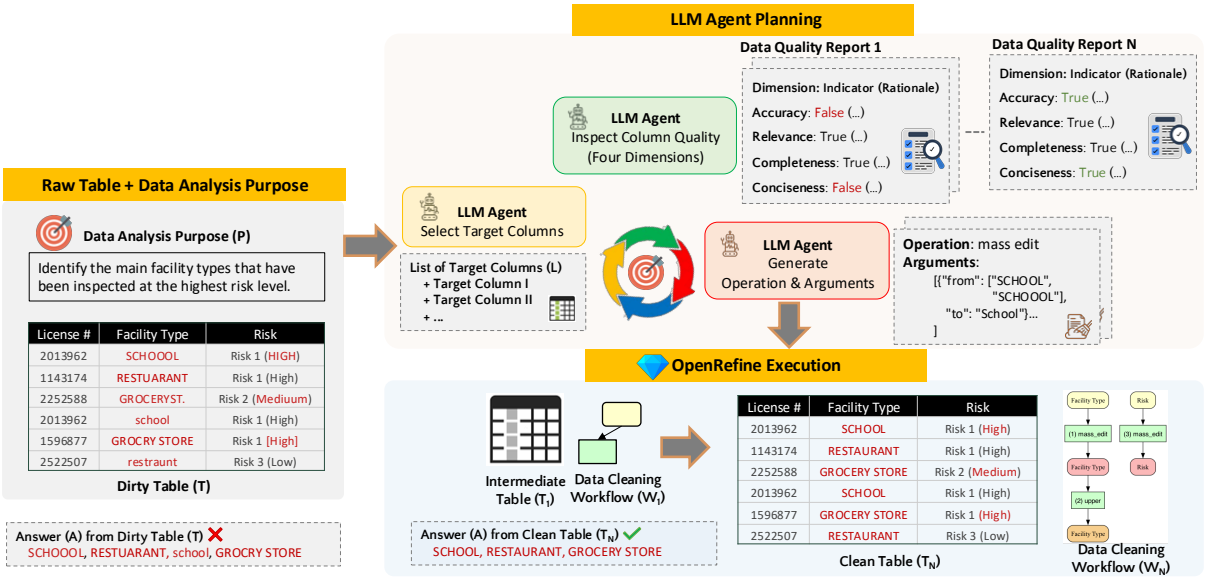

Figure 2: Architecture of the AutoDCWorkflow framework. Given a data analysis purpose $\mathbf{P}$: Identify the main facility types that have been inspected at the highest risk level, and a dirty table $\mathbf{T}$, we first run the purpose on $\mathbf{T}$ and obtain an incorrect answer $\mathbf{A}$: "SCHOOOL, RESTUARANT, school, GROCRY STORE". An iterative data cleaning process is then initiated, consisting of: (1) *Selecting Target Columns*, (2) *Inspecting Column Quality*, (3) *Generating Operations & Arguments*, and (4) *Executing Operations* via the OpenRefine API. In each iteration, the framework predicts and applies the next operation, evaluates the resulting column quality, and continues until all target columns satisfy the required quality standards (i.e., all dimensions are marked as `True` in the Data Quality Report). This process yields a cleaned table $\mathbf{T}_N$ and a corresponding complete workflow $\mathbf{W}_N$. Running the purpose on $\mathbf{T}_N$ produces the correct answer $\mathbf{A}$: "SCHOOL, RESTAURANT, GROCERY STORE".

port data cleaning. Holoclean (Rekatsinas et al., 2017), for example, leverages probabilistic inference to repair data by modeling integrity constraints and correlations. Pre-trained language models (PLMs) have been applied to entity matching (EM) tasks (Kasai et al., 2019; Li et al., 2020; Suhara et al., 2022; Fang et al., 2023; Wadhwa et al., 2024), with a primary focus on duplicate detection and resolution. Recent work has explored the use of large language models (LLMs) to automate various data cleaning tasks. RetClean (Eltabakh et al., 2024) applies retrieval-augmented generation with LLMs to impute missing values and correct erroneous entries. LLMClean (Biester et al., 2024) uses LLMs to automatically construct context models—specifically, Ontological Functional Dependencies (OFDs)—to detect and repair inconsistencies in relational data. CleanAgent (Qi and Wang, 2024) combines LLMs with a declarative API to support standardization tasks such as date formatting and categorical value normalization. While these approaches demonstrate the potential of LLMs in individual cleaning tasks, they do not generate end-to-end, executable workflows tailored to specific analytical purposes.

LLMs for Tabular Understanding and Transformation. Large language models (LLMs) have been applied to various tabular tasks, including fact verification (Chen et al., 2019; Fang et al., 2025a) and question answering (Jin et al., 2022; Zou et al., 2025), and structured reasoning (Cheng et al., 2023; Fu et al., 2024). CHAIN-OF-TABLE (Wang et al., 2024b) demonstrated a step-by-step approach in which LLMs simulate reasoning over table transformations via synthetic operations, highlighting the potential for structured tabular problem-solving. Gandhi et al. (Gandhi et al., 2024) further showed that clear task descriptions can guide one-shot transformation plans, underscoring the role of instruction design in dataset repurposing.

3 Background

3.1 Purpose-Driven Data Cleaning Workflows

Our approach is motivated by the principle that "*high-quality data is fit-for-purpose*", emphasizing that each data cleaning task should be driven by a well-defined data analysis purpose (Paul Bédard and Barnes, 2010; Sidi et al., 2012; Burden and Stenberg, 2024). Accordingly, different purposes may produce distinct result tables and necessitate different data cleaning operations. The data analysis purposes in our benchmark are summarized in Table 2, categorized and documented as follows:

Descriptive Statistics: This involves general statistics or aggregates about the dataset. For example,

Category	Number of Purposes
Descriptive Statistics	26
Counting and Grouping	27
Category or Type Classification	19
Time-based Analysis	17
Correlations and Relationships	10
Filtering and Specific Queries	43

Table 2: Summary of Data Analysis Purposes Categories.

we might ask, "What's the highest or lowest loan amount for each zip code?"

Counting and Grouping: Identifying distinct elements or counting occurrences within the dataset, e.g., "Count how many types of risks are recorded in the dataset."

Category or Type Classification: Classifying data based on certain criteria. For instance, "Identify which sponsors provide breakfast."

Time-based Analysis: Analyzing trends or patterns over time. For example, "Find dishes that first appeared before the year 2000."

Correlations and Relationships: Exploring relationships between different columns in the dataset. For instance, "Examine if a correlation exists between jobs reported and the loan amount received."

Filtering and Specific Queries: Performing specific queries or filtering the dataset based on defined conditions, e.g., "Report all NAICS Codes that indicate job counts greater than 3.0."

3.2 OpenRefine Workflow for Transparency

OpenRefine (Guidry, 2020) enhances transparency in data cleaning by documenting operations within a workflow, enabling traceability of intermediate tables transformed by each step. The OpenRefine workflow records how the initial dataset D_0 evolves into the final version D_n through a sequence of operations $\mathbf{O}_1, \ldots, \mathbf{O}_n$ (Li et al., 2021):

$$D_0 \overset{\mathbf{O}_1}{\rightsquigarrow} D_1 \overset{\mathbf{O}_2}{\rightsquigarrow} D_2 \overset{\mathbf{O}_3}{\rightsquigarrow} \cdots \overset{\mathbf{O}_n}{\rightsquigarrow} D_n \,. \quad (1)$$

A workflow includes (1) data operations $\mathbf{O}_i$ applied to transform the data and (2) intermediate tables D_i updated at each step.

4 Approach Architecture

The complete AutoDCWorkflow pipeline, shown in Figure 2, consists of two main stages: (1) LLM agent planning and (2) data cleaning via OpenRefine execution.

Problem Definition. Given a table $\mathbf{T}$ and a data analysis purpose $\mathbf{P}$, AutoDCWorkflow aims to generate data cleaning workflow $\mathbf{W}_N$ and the cleaned table $\mathbf{T}_N$. To achieve this, AutoDCWorkflow begins by identifying a list of target columns $\mathbf{L}$. Then it iteratively selects and cleans columns and removes them from $\mathbf{L}$ once successfully cleaned. In the OpenRefine stage, the generated operations and arguments $\mathbf{O}(\cdot)$ are executed to modify the input table into an intermediate table $\mathbf{T}_i$. Each operation $\mathbf{O}_i(\cdot)$ is logged in the data cleaning workflow $\mathbf{W}_i$. The updated table $\mathbf{T}_i$ and workflow $\mathbf{W}_i$ then serve as inputs for the next LLM planning iteration. This process continues until $\mathbf{L}$ is empty, indicating that all target columns have been cleaned in alignment with $\mathbf{P}$. The final cleaned table $\mathbf{T}_N$, produced by the workflow $\mathbf{W}_N$ as $\mathbf{T}_N = \mathbf{W}_N \circ \mathbf{T}$, is claimed to be of high quality and capable of generating the correct answer $\mathbf{A}$ for the purpose $\mathbf{P}$.

4.1 Select Target Columns

Purpose-driven data cleaning emphasizes data quality fit-for-purpose (Paul Bédard and Barnes, 2010; Sidi et al., 2012; Burden and Stenberg, 2024). Not all columns are necessarily relevant to the purpose of the given analysis. Select Target Columns agent aims to reduce the complexity of data cleaning from the entire table to only the columns pertinent to the given purpose (Gandhi et al., 2024).

We design the prompt template with task instructions and few-shot demonstration examples, inspired by Gandhi et al. (2024). Each example contains the following content: (1) table information, (2) purpose statement, (3) explanation or rationale for identifying relevant columns based on the purpose statement, and (4) target columns.

4.2 Inspect Column Quality

Data quality is a multidimensional concept (Wand and Wang, 1996). Wang et al. (2024a) identify six key dimensions: *completeness*, *accuracy*, *timeliness*, *consistency*, *relevance*, and *concise representation*, while emphasizing that the choice of dimensions often depends on the study context. In our setting, we exclude *timeliness*, as no timestamps or external knowledge are available to evaluate data freshness, and *consistency*, which is typically defined through rule-based dependencies across multiple columns, is not feasible in OpenRefine's operation model that emphasizes single-column transformations. Accordingly, this paper focuses on four widely adopted dimensions: *accuracy*, *relevance*, *completeness*, and *conciseness*.

The Inspect Column Quality agent produces a Data Quality Report assessing these four quality

Operation	Description	Quality Dimensions
`upper`	Converts text to uppercase for consistent casing.	Conciseness
`trim`	Removes leading/trailing spaces.	Accuracy, Conciseness
`numeric`	Converts values to numeric format if applicable.	Accuracy
`date`	Standardizes date format across entries.	Accuracy
`mass_edit`	Merges similar or erroneous values.	Accuracy, Conciseness, Relevance, Completeness
`regexr_transform`	Uses regex to match and replace patterns.	Accuracy, Conciseness, Relevance

Table 3: Mechanisms and quality dimensions of common data operations.

dimensions, defined as follows: *Accuracy* evaluates whether the target column is free from obvious errors, inconsistencies, or biases. *Relevance* determines if the target column is essential to addressing the analysis objectives. *Completeness* checks if the target column has an adequate sample size with minimal missing values. *Conciseness* assesses if spellings are standardized and whether there are no different representations for the same concept.

It is important to note that *relevance* is implicitly assessed by the Select Target Column agent and explicitly reevaluated by the Inspect Column Quality agent. This reevaluation ensures that the values in the selected column remain aligned with the data analysis purpose, thereby reducing the risk of misalignment introduced in the earlier stage.

Data cleaning objectives are specific and focused on addressing errors in the target column. They are generated based on the Data Quality Report if any of the four quality dimensions are evaluated as *False*. This process guides the prediction of the next operation and its arguments.

4.3 Generate Operation and Arguments

Instructions and example usages of data cleaning operations serve as prompts to guide AutoDCWorkflow in generating appropriate operations and corresponding arguments iteratively. We include six commonly used OpenRefine operations: `upper`, `trim`, `numeric`, `date`, `mass_edit`, and `regexr_transform`. The mechanisms of these operations and their associated data quality dimensions are summarized in Table 3. Crucially, a single operation can impact multiple data quality dimensions. Despite the limited number of operations, the vast space of possible arguments and the flexible combination of operations enable our approach to handle complex real-world data quality issues.

Operation Generation. We design a structured prompt template (see Appendix A.7). This template incorporates task instructions, operation definitions, examples, and explanations. The LLM agent determines the operation that best fits the given table contents and data analysis purpose.

Arguments Generation. The `mass_edit` and `regexr_transform` operations require argument generation to handle various data errors (see Appendix A.3). We provide few-shot examples demonstrating arguments usage for specific data cleaning objectives.

Dataset	# Purposes	# Answers	# Tables	# Workflows	# Rows	# Columns
Menu	30	30	3	30	100	20
Dish	16	16	9	16	50	8
Flights	16	16	16	16	50	7
PPP	22	22	11	22	20	14
Hospital	28	28	28	28	20	12
CFI	30	30	29	30	10	11
Total	142	142	96	142	-	-

Table 4: Summary of Benchmark Dataset Statistics. Total integrates six datasets. Each purpose is assigned to a single table, and the # Tables represents the number of distinct tables in each dataset.

5 Benchmark Demonstration

The benchmark includes annotated datasets and evaluation dimensions to assess workflow quality. The benchmark dataset comprises data analysis purposes, answer sets, raw tables[1], and manually curated tables and workflows. The statistic of the benchmark dataset is provided in Table 4. The benchmark evaluation enables a robust assessment of generated workflows in three key dimensions:

[RQ1] Purpose Answer Dimension: Can we retrieve approximately or exactly the same correct answer from the repaired clean table?

[RQ2] Column Value Dimension: How closely does the repaired table resemble the human-curated table?

[RQ3] Workflow (Operation) Dimension: To what extent does AutoDCWorkflow generate the correct and complete set of operations?

[1]The number of tables does not match the number of purposes in Menu, Dish, PPP, and CFI because multiple analysis purposes were performed on the same initial tables.

5.1 Dataset Construction

5.1.1 Dataset Preparation and Partition

The benchmark spans multiple domains, including Menu and Dish[2] (social science), Chicago Food Inspection (CFI)[3] and Hospital (public health) (Mahdavi and Abedjan, 2019), Paycheck Protection Program (PPP)[4] (finance), and Flights (transportation) (Mahdavi and Abedjan, 2019). To evaluate the LLM's ability to generate workflows on datasets with diverse data types and varying levels of data quality issues, we sample tables from these original datasets and systematically inject different types of errors based on their associated analysis purposes.

We implement Python functions to inject common real-world data quality issues into clean datasets. These include: (1). Duplicate variants: Slightly different but semantically equivalent values (e.g., "McDonalds" vs. "Mc Donald's"). (2). Formatting inconsistencies: Extra whitespace and non-ASCII characters. (3). Case variations: Mixed use of uppercase and lowercase letters. (4). Type errors: Inserting strings into numeric fields (e.g., "N/A" in a price column). These errors simulate real-world data quality challenges, allowing us to assess the model's robustness in data cleaning. The error injection rate can be found in Table 5. Moreover, empty columns are removed, as they do not contribute meaningful information for answering the given purposes.

5.1.2 Purpose and Answer Annotation

AutoDCWorkflow benchmark defines 142 purposes based on the sample rows and column schema (see Table 6). These purposes are constructed using two methods: referencing columns by exact or paraphrased names and specifying specific data values. The first method allows LLMs to recognize and identify target columns, while the second adds complexity by requiring inference of relationships between cell values and the column schema.

To assess the difficulty of each purpose, we consider two factors: (1) the number of target columns involved and (2) the diversity of their data types. We hypothesize that purposes requiring more target columns and diverse data types provide a more challenging task for LLMs to generate workflows. For instance, PPP tables primarily contain numerical data with less variation than other datasets, making them a relatively simpler task for LLMs.

Dataset	# Tables	Error Ratio (mean ± std)
Menu	30	0.0636 ± 0.1078
Dish	16	0.1480 ± 0.0745
Flights	16	0.2584 ± 0.0942
PPP	22	0.3451 ± 0.1930
Hospital	28	0.1463 ± 0.0498
CFI	30	0.2307 ± 0.0545

Table 5: Error injection rates across tables.

Answer Annotation. Querying the dataset returns purpose answers in three forms: a single numeric value, a string, or a collection of columns and cell values. This enables us to compare the purpose answers derived from LLM-generated tables with those from ground truth tables.

Dataset	# Purposes	# Target Columns			Target Column Types		
		1	2	≥ 3	Numeric	Text	Date
Menu	30	16	13	1	11	23	2
Dish	16	1	6	9	11	15	5
Flights	16	4	8	4	0	6	13
PPP	22	4	16	2	21	13	0
Hospital	28	3	15	10	6	28	0
CFI	30	10	16	4	4	29	4

Table 6: Summary of Data Analysis Purposes Statistics. Each row represents the number of target columns and column types per purpose, providing insights into the complexity of data analysis purposes across different datasets.

5.1.3 Workflow Annotation

Different human curators may prefer different data cleaning operations, leading to varied workflows for the same data cleaning task. As a result, there is no single gold-standard workflow. In this study, a data cleaning expert utilized OpenRefine operations, cleaning the given table for a data analysis purpose and annotated data cleaning workflows as a reference, which we regard as *silver-standard* ground truth for evaluating LLM-generated workflows. We assess recall and accuracy by comparing the operations predicted by LLMs to those annotated in the *silver* workflows. This comparison assesses how closely LLMs approximate the silver-standard ground truth rather than serve as a strict indicator of workflow correctness.

Since datasets differ in data types and the distribution of quality issues, each dataset has its own most frequently used operations. For instance, *numeric* is the most frequently used operation in PPP, while *date* is the most frequently used in Flights, as shown in Figure 3.

[2] https://menus.nypl.org/

[3] https://data.cityofchicago.org/Health-Human-Services/Food-Inspections/

[4] https://data.sba.gov/dataset/ppp-foia/

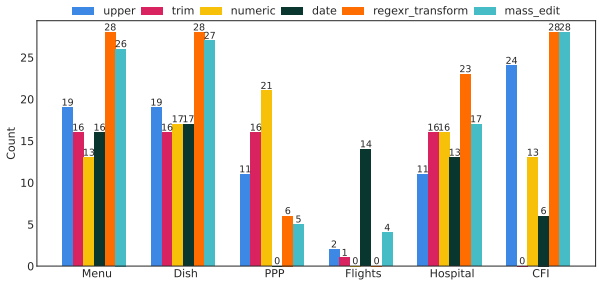

Figure 3: Frequency distribution of data operation types across all datasets.

5.2 Evaluation Dimensions and Metrics

We assess the generated workflows across three key dimensions: (1) **Accuracy**, by evaluating whether the cleaned table produces the correct answer for the given purpose (RQ1 Answer Dimension); (2) **Consistency**, by measuring how closely the repaired table aligns with the ground truth (RQ2 Column Value Dimension); and (3) **Plausibility**, determining the extent to which the generated workflow resembles the human-curated ground truth (RQ3 Workflow Dimension).

5.2.1 Purpose Answer Dimension

The purpose answer dimension assesses the accuracy of the workflow by measuring how well the cleaned table produces correct answers aligned with the ground truth. We evaluate alignment using *Precision*, *Recall*, *F1*, and *Similarity*. Exact matches are used for single-valued answers (e.g., numbers or strings), while complex answers (e.g., lists or tables in JSON format) are assessed element-wise with *Precision*, *Recall*, and *F1* scores. The *Similarity* quantifies the proportion of matching characters relative to the total number of characters across both sequences. This ensures a comprehensive evaluation of workflow accuracy across different answer types and complexities. The matching characters are captured by finding the longest contiguous matching block, recursively applying the same logic to the unmatched regions to the left and right of the match, and summing all matching blocks to compute total matches.

5.2.2 Column Value Dimension

This dimension assesses the consistency of data cleaning workflows by measuring how closely cleaned column values align with the ground truth table. The column cleanness ratio represents the average proportion of matching values across all target columns. Equivalence comparison is performed based on data types: numeric, string, and date. The detailed formula for calculating the column cleanness ratio is shown in Equation 2.

$$\text{Ratio} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N}\delta(T_{i,j}, G_{i,j}) \qquad (2)$$

where: M: Total number of target columns, N: Total number of rows, $T_{i,j}$: Value in row i and column j of the table, $G_{i,j}$: Value in row i and column j of the ground truth, $\delta(T_{i,j}, G_{i,j})$:

$$\begin{cases} 1, & \text{if } T_{i,j} \text{ is numeric or } \text{lower}(T_{i,j}) = \text{lower}(G_{i,j}) \\ 0, & \text{otherwise} \end{cases}$$

5.2.3 Workflow (Operation) Dimension

To assess the plausibility of generated workflows, this dimension evaluates how well LLM-generated operations align with the correct operations in the *silver* workflows. We quantify this alignment using *Precision*, *Recall*, and *F1-score*, which measure the degree of overlap between operations in both workflows. Discrepancies between LLM-generated and *silver* workflows highlight how models differ in the selection of operations resolving the same data quality issues. These variations reflect the model's ability to apply different yet valid approaches to achieve the same data cleaning objective, showcasing a distinct "flavor" of workflows while still fulfilling the given purpose.

6 Experiments

6.1 Experiment Settings

We use raw table inputs (without any prior data cleaning) and a direct prompting (DP) approach as the baseline. In the DP setting, the LLM is instructed to generate the entire data cleaning workflow in a single step. Unlike the iterative pipeline in AutoDCWorkflow, DP receives the same context—including the list of available operations and example workflows—but lacks access to intermediate reasoning stages and multi-turn decision-making. We evaluate AutoDCWorkflow using three widely adopted, open-access LLMs of comparable model sizes: Llama 3.1-8B (Dubey et al., 2024), Gemma 2-9B (Rivière et al., 2024), and Mistral-7B (Jiang et al., 2023). To assess the impact of model size, we additionally include Gemma 2-27B (Rivière et al., 2024) in our comparison. This allows us to investigate how scaling model size influences workflow generation. For operation prediction, the models are configured with a

Model	Answer Dimension				Column Dimension	Workflow Dimension			
	Precision	Recall	F1	Similarity	Ratio	Exact	Precision	Recall	F1
Baseline (Raw Tables)	0.2345	0.2375	0.2201	0.4678	0.4262	–	–	–	–
Llama 3.1–8B (DP)	0.3070	0.2914	0.2811	0.5039	0.5817	0.0493	0.3768	0.1554	0.2053
Llama 3.1–8B (AutoDCWorkflow)	0.5415^{**++}	0.5185^{**++}	0.5181^{**++}	0.6868^{**++}	0.7475^{**++}	0.0986	**0.9331**$^{++}$	0.5265 $^{++}$	0.6447^{++}
Mistral–7B (DP)	0.2862	0.2740	0.2612	0.4933	0.4884	0.0000	0.1162	0.0363	0.0536
Mistral-7B (AutoDCWorkflow)	0.3635**	0.3320*	0.3320*	0.5208	0.6004^{**++}	0.0493^{++}	0.7656^{++}	0.4096^{++}	0.5065^{++}
Gemma2–9B (DP)	0.3368	0.3098	0.3030	0.5109	0.6032	0.0141	0.4131	0.1438	0.2042
Gemma2–9B (AutoDCWorkflow)	0.3841^{**++}	0.3502^{**+}	0.3479^{**++}	0.5543^{+}	0.6575^{**++}	**0.1056**$^{++}$	0.8853 $^{++}$	**0.5698**$^{++}$	**0.6640**$^{++}$
Gemma2–27B (DP)	0.3298	0.3039	0.2956	0.5073	0.5967	0.0141	0.2887	0.0937	0.1353
Gemma2–27B (AutoDCWorkflow)	**0.6590**$^{**++}$	**0.6234**$^{**++}$	**0.6256**$^{**++}$	**0.7567**$^{**++}$	**0.7900**$^{**++}$	0.0775^{+}	0.8759^{++}	0.5514^{++}	0.6456^{++}

Table 7: Overall performance on the combined dataset, including PPP, CFI, Dish, Menu, Hospital, and Flights. Baseline (Raw Table) uses raw tables without cleaning. DP refers to directly using a single prompt for LLMs to generate appropriate data cleaning operations. **Bold** indicates the highest result in each column. AutoDCWorkflow results are compared with baseline (*) and DP (+) with statistical paired t-test. **(*) represents a 99% (95%) confidence level compared with baseline (Raw Table). ++(+) represent a 99% (95%) confidence level compared with direct prompting (DP).

default *temperature* of 0.1, which is increased to 0.3 if no valid operation is returned (see more hyperparameter settings in Appendix A.5). The total number of tokens for all input prompts, excluding table data, is approximately 4538 tokens. The maximum number of output tokens is also set to 2048, with the stop word defined as ["\n\n\n"]. The experiment is conducted using four Nvidia A100 GPUs via the NCSA Delta system via ACCESS program (Boerner et al., 2023). Our implementation code is available in an anonymous repository[5].

6.2 Experiment Results

We compare AutoDCWorkflow against two baselines: (1) raw tables, and (2) Direct Prompting (DP). As shown in Table 7, AutoDCWorkflow outperforms both baselines across all evaluation dimensions. A detailed performance table of six datasets can be found in Appendix A.6.

Comparison with Baseline for Comparable Models. In the Answer Dimension, AutoDCWorkflow consistently outperforms both baselines in precision, recall, and F1. Notably, Llama 3.1–8B and Gemma 2–9B achieve statistically significant improvements in F1 over both baselines at the 99% confidence level. For instance, Llama 3.1-8B improves from 0.2201 (Raw Table) and 0.2811 (DP) to 0.5181, and Gemma 2–9B from 0.3030 (DP) to 0.3479.

In the Column Dimension, all models using AutoDCWorkflow significantly improve the column value matching ratio compared to both baselines. For example, Llama 3.1–8B rises from 0.4262 (raw) and 0.5817 (DP) to 0.7475, while Gemma 2-9B reaches 0.6575. Even Mistral–7B, the smallest model, shows a strong gain (from 0.4884 to 0.6004), all statistically significant at the 99% level.

In the Workflow Dimension, AutoDCWorkflow substantially improves precision, recall, and F1 across all models relative to DP, again with 99% confidence. These results highlight the effectiveness of our iterative pipeline in producing higher-quality, purpose-aligned workflows, addressing the limitations of direct prompting.

Performance of Large-Scale Model (Gemma 2–27B). Among all models, Gemma 2–27B with AutoDCWorkflow achieves the best performance across nearly every dimension. It reaches the highest F1 score in the Answer Dimension (0.6256), the best column matching ratio (0.7900), and strong workflow metrics (precision 0.8759, F1 0.6456), all statistically significant at the 99% confidence level. These results suggest that the larger model (27B) benefits more from the iterative, purpose-driven cleaning process. This finding indicates exploring a potential scaling law in data cleaning workflow generation, whether increasing model size leads to better reasoning, quality inspection, and workflow generation. Investigating this relationship would be a promising direction for future work.

7 Case Study

In this section, we further analyze how different models generate varying workflows by examining their outputs. We showcase how Llama 3.1-8B and Gemma 2-9B construct workflows with different operations while successfully repairing data to produce the correct answer. The analysis focuses on the specific purpose of identifying the facility type inspected multiple times on the most recent date.

[5] https://github.com/LanLi2017/LLM4DC

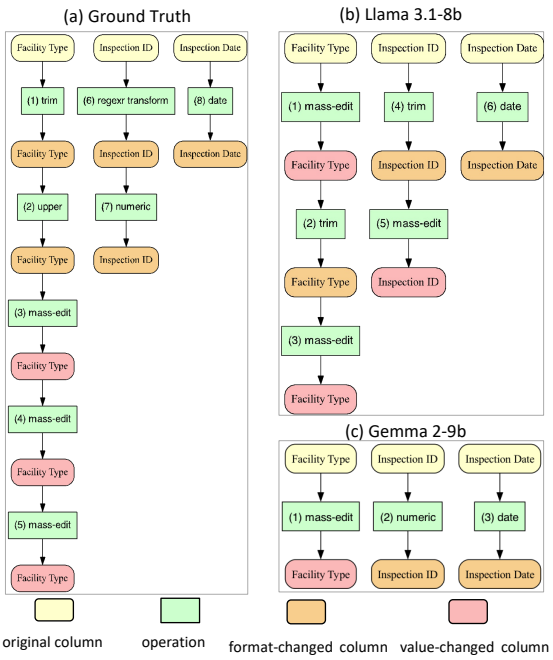

Figure 4: Three workflow versions generated by (a) Ground Truth (human curators), (b) Llama 3.1-8B, and (c) Gemma 2-9B.

Figure 4 visualizes the workflows using ORMA[6], where each workflow is modeled as a graph with *data nodes* (colorful rounded rectangles representing columns) and *operation nodes* (green boxes representing applied operations) (Li et al., 2021). The three branches in each diagram correspond to the target columns—Facility Type, Inspection ID, and Inspection Date—allowing a direct comparison of how each model processes them.

Facility Type Column: The ground truth workflow applies *trim* (1x), *upper* (1x), and *mass_edit* (3x). Llama 3.1-8b correctly selects *trim* (1x) and *mass_edit* (2x), while Gemma 2-9b consolidates all repairs into a single *mass_edit* operation. Inspection ID Column: The ground truth workflow uses *regexr_transform* and *numeric* to standardize non-numeric IDs. Llama 3.1-8B uses *mass_edit* to accomplish the same result. Inspection Date Column: Both models detect date-time inconsistencies and apply the *date* operation for correction. Despite using fewer steps and varying approaches, both Llama 3.1-8B and Gemma 2-9B effectively clean the table and produce the correct purpose answer: "RESTAURANT". This demonstrates their ability to generate accurate workflows efficiently.

[6]Arrows indicate the flow of data cleaning operations across specific columns.

8 Conclusions

This paper presents AutoDCWorkflow, a framework to automatically generate data cleaning workflows using LLMs, supported by a new benchmark and evaluation metrics. Our experiments across 96 datasets and 142 tasks show that all LLM backbone models are capable of interpreting the given purpose and reasoning about appropriate data cleaning operations to resolve data quality issues. Gemma 2-27B yields the most accurate cleaned tables and downstream answers, while Gemma 2-9B produces workflows most consistent with human annotations. These findings underscore that different LLMs capture complementary aspects of data cleaning, making model choice an important consideration for different objectives. Beyond performance comparisons, our iterative design demonstrates that LLMs can reason about and adapt to diverse data quality issues, moving beyond one-off cleaning heuristics. Taken together, this work establishes a foundation for reproducible evaluation and highlights the broader potential of LLM-driven approaches to generalize across domains, error types, and data cleaning pipelines. Looking forward, we envision general-purpose LLM agents that support end-to-end data cleaning, making structured data more transparent, reusable, and analysis-ready.

Limitations

While our study reveals several limitations of AutoDCWorkflow, we also emphasize its potential to generalize beyond the current experimental scope and aim to draw interest in LLMs for data cleaning.

Handling High Data Noise. When the data is highly noisy, LLMs may struggle to learn from the provided examples. Beyond simply adding more in-context examples, incorporating metadata or domain-specific knowledge could improve repair accuracy. Another direction is self-evolving prompting, where demonstration examples are refined based on prior outputs to iteratively enhance cleaning performance.

Single-Column Repairing. AutoDCWorkflow processes tables column by column independently, ignoring correlations and functional dependencies across columns. This limits contextual evidence for LLMs and can reduce the accuracy of generated operations. Extending the framework with multi-column reasoning would enable it to handle more complex data cleaning issues involving inter-column relationships, as a future direction.

Computational Resources. Experiments were limited to Gemma 2–27B due to computational resource constraints. Evaluating larger models may further improve cleaning accuracy and workflow quality.

Sampling Bias. Sampling a subset of column values may overlook certain error types, particularly rare ones. To address this, our system applies iterative sampling for large columns until the LLM-generated data quality report rates all dimensions as "good" across multiple rounds, serving as a heuristic for coverage.

Ethics Statement

rAll datasets are publicly available, properly licensed, and free of personally identifiable information. AutoDCWorkflow employs LLMs to generate workflows as inference outputs, which are compared against human-annotated ground truth. We release benchmark resources openly to promote reproducibility and responsible research.

Acknowledgements

This work used Delta advanced computing and data resources, supported by the National Science Foundation (award OAC 2005572) and the State of Illinois, through allocation CIS230333 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program. ACCESS is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296. Delta is a joint effort of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications.

References

Fabian Biester, Mohamed Abdelaal, and Daniel Del Gaudio. 2024. LLMClean: Context-aware tabular data cleaning via llm-generated ofds. In *New Trends in Database and Information Systems - ADBIS 2024 Short Papers, Workshops, Doctoral Consortium and Tutorials*.

Timothy J. Boerner, Stephen Deems, Thomas R. Furlani, Shelley L. Knuth, and John Towns. 2023. ACCESS: advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing, PEARC 2023, Portland, OR, USA, July 23-27, 2023*, pages 173–176. ACM.

Håkan Burden and Susanne Stenberg. 2024. Fit for purpose–data quality for artificial intelligence.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*.

Zui Chen, Lei Cao, Sam Madden, Tim Kraska, Zeyuan Shang, Ju Fan, Nan Tang, Zihui Gu, Chunwei Liu, and Michael Cafarella. 2023. Seed: Domain-specific data curation with large language models. *arXiv e-prints*, pages arXiv–2310.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Binding language models in symbolic languages. *ICLR*.

Xu Chu, Ihab F Ilyas, and Paolo Papotti. 2013. Holistic data cleaning: Putting violations into context. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 458–469. IEEE.

Michele Dallachiesa, Amr Ebaid, Ahmed Eldawy, Ahmed Elmagarmid, Ihab F Ilyas, Mourad Ouzzani, and Nan Tang. 2013. Nadeef: a commodity data cleaning system. In *Proceedings of the 2013 ACM SIGMOD*, pages 541–552.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Mohamed Y. Eltabakh, Zan Ahmad Naeem, Mohammad Shahmeer Ahmad, Mourad Ouzzani, and Nan Tang. 2024. Retclean: Retrieval-based tabular data cleaning using llms and data lakes. *Proc. VLDB Endow.*, 17(12):4421–4424.

Liri Fang, Dongqi Fu, and Vetle I Torvik. 2025a. Automatic scientific claims verification with pruned evidence graph. In *ICLR 2025 Agentic AI Workshop*.

Liri Fang, Lan Li, Yiren Liu, Vetle I. Torvik, and Bertram Ludäscher. 2023. KAER: A knowledge augmented pre-trained language model for entity resolution. *Knowledge Augmented Methods for Natural Language Processing workshop in conjunction with AAAI 2023*.

Liri Fang, Malik Oyewale Salami, Griffin M. Weber, and Vetle I. Torvik. 2025b. ucite: The union of nine large-scale public pubmed citation datasets with reliability filtering. *Data in Brief*, 60:111535.

Dongqi Fu, Liri Fang, Zihao Li, Hanghang Tong, Vetle I. Torvik, and Jingrui He. 2024. Parametric graph representations in the era of foundation models: A survey and position. *CoRR*, abs/2410.12126.

Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. 2024. Better synthetic data by retrieving and transforming existing datasets. In *Findings of ACL*.

Thad Guidry. 2020. OpenRefine server side architecture. Accessed September 2025.

Albert Q. Jiang, Alexandre Sablayrolles, and etc. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Nengzheng Jin, Joanna Siebert, Dongfang Li, and Qingcai Chen. 2022. A survey on table question answering: recent advances. In *China Conference on Knowledge Graph and Semantic Computing*, pages 174–186. Springer.

Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 3363–3372.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource deep entity resolution with transfer and active learning. In *ACL*.

Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael J Franklin, and Ken Goldberg. 2016. Activeclean: Interactive data cleaning for statistical modeling. *Proceedings of the VLDB Endowment*, 9(12):948–959.

Lan Li and Bertram Ludäscher. 2023. On the reusability of data cleaning workflows. *International Journal of Digital Curation*, 17(1):6.

Lan Li, Nikolaus Nova Parulian, and Bertram Ludäscher. 2021. Automatic module detection in data cleaning workflows: Enabling transparency and recipe reuse. *International Journal of Digital Curation*, 16(1):11.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment*, 14(1):50–60.

Mohammad Mahdavi and Ziawasch Abedjan. 2019. Reds: Estimating the performance of error detection strategies based on dirtiness profiles. In *Proceedings of the 31st International Conference on Scientific and Statistical Database Management*, pages 193–196.

Timothy McPhillips, Craig Willis, Michael R Gryk, Santiago Nunez-Corrales, and Bertram Ludäscher. 2019. Reproducibility by other means: Transparent research objects. In *2019 15th International Conference on EScience (EScience)*, pages 502–509. IEEE.

Sedir Mohammed, Felix Naumann, and Hazar Harmouch. 2025. Step-by-step data cleaning recommendations to improve ML prediction accuracy. In *Proceedings 28th International Conference on Extending Database Technology, EDBT 2025*, pages 542–554.

Nikolaus Nova Parulian and Bertram Ludäscher. 2022. DCM explorer: a tool to support transparent data cleaning through provenance exploration. In *Proceedings of the 14th International Workshop on the Theory and Practice of Provenance, TaPP 2022*, pages 10:1–10:6.

L Paul Bédard and Sarah-Jane Barnes. 2010. How fit are your data? *Geostandards and Geoanalytical Research*, 34(3):275–280.

Leo Pipino, Yang W. Lee, and Richard Y. Wang. 2002. Data quality assessment. *Commun. ACM*, 45(4):211–218.

Danrui Qi and Jiannan Wang. 2024. Cleanagent: Automating data standardization with llm-based agents. *CoRR*, abs/2403.08291.

Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, and Christopher Ré. 2017. Holoclean: Holistic data repairs with probabilistic inference. *Proc. VLDB Endow.*, 10(11):1190–1201.

El Kindi Rezig, Lei Cao, Michael Stonebraker, Giovanni Simonini, Wenbo Tao, Samuel Madden, Mourad Ouzzani, Nan Tang, and Ahmed K Elmagarmid. 2019. Data civilizer 2.0: A holistic framework for data preparation and analytics. *Proceedings of the VLDB Endowment*, 12(12):1954–1957.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, and etc. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

Shafaq Siddiqi, Roman Kern, and Matthias Boehm. 2023. Saga: a scalable framework for optimizing data cleaning pipelines for machine learning applications. *Proceedings of the ACM on Management of Data*, 1(3):1–26.

Fatimah Sidi, Payam Hassany Shariat Panahy, Lilly Suriani Affendey, Marzanah A Jabar, Hamidah Ibrahim, and Aida Mustapha. 2012. Data quality: A survey of data quality dimensions. In *2012 International Conference on Information Retrieval & Knowledge Management*, pages 300–304. IEEE.

Michael Stonebraker, Ihab F Ilyas, et al. 2018. Data integration: The current status and the way forward. *IEEE Data Eng. Bull.*, 41(2):3–9.

Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating columns with pre-trained language models. In *Proceedings of the 2022 International Conference on Management of Data*.

Somin Wadhwa, Adit Krishnan, Runhui Wang, Byron C Wallace, and Luyang Kong. 2024. Learning from natural language explanations for generalizable entity matching. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Yair Wand and Richard Y Wang. 1996. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11):86–95.

Jingran Wang, Yi Liu, Peigong Li, Zhenxing Lin, Stavros Sindakis, and Sakshi Aggarwal. 2024a. Overview of data quality: Examining the dimensions, antecedents, and impacts of data quality. *Journal of the Knowledge Economy*, 15:1159–1178.

Richard Y. Wang and Diane M. Strong. 1996. Beyond accuracy: What data quality means to data consumers. *J. Manag. Inf. Syst.*, 12(4):5–33.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. 2024b. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *The Twelfth International Conference on Learning Representations, ICLR 2024*.

Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, and etc. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3:160018.

Jiaru Zou, Dongqi Fu, Sirui Chen, Xinrui He, Zihao Li, Yada Zhu, Jiawei Han, and Jingrui He. 2025. GTR: graph-table-rag for cross-table question answering. *CoRR*, abs/2504.01346.

A Appendix

A.1 OpenRefine Workflow Demonstration

Unlike script-based data cleaning, OpenRefine offers a more transparent approach by explicitly documenting data cleaning operations within a workflow, enabling traceability of intermediate tables transformed by each operation.

OpenRefine records the workflow in a JSON format file, which includes detailed information about each operation, such as the operation name, functions, and other relevant parameters (see Figure 5). This workflow provides a clear and reproducible record of how the initial dataset is transformed into its final version through a sequence of data cleaning operations.

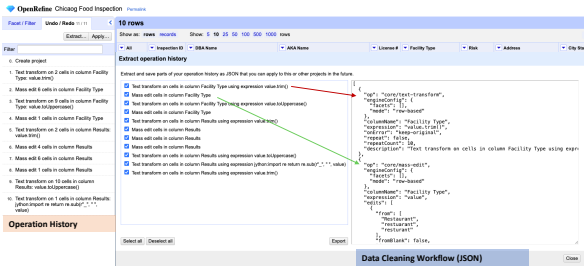

Figure 5: As users interactively manipulate the dataset, OpenRefine automatically records each operation into the *Operation History* panel (left). These operations can then be exported as a structured *Data Cleaning Workflow* in JSON format (right), enabling reproducibility and reuse across projects.

A.2 Dataset Description

We include six datasets across six topics in our benchmark: Menu, Dish, Chicago Food Inspection (CFI) data, Paycheck Protection Program (PPP) loan data, Hospital and Flights datasets.

Every row of the Menu table represents a menu record, capturing both "external" and "internal" information. For instance, the column *event* records the "external" information regarding the meal types associated with the menu, such as *dinner*, *lunch*, breakfast, or other special events. Meanwhile, the column *page_count* represents the "internal" information, documenting the number of pages for this menu. Records in the Dish table capture details about dishes featured on menus, including dish names, frequency of appearance, first and last years on the menu, and highest and lowest prices. This information helps analyze trends in dish popularity over time.

For every row in the CFI dataset sample tables, the data represents inspection results for restaurants and food facilities across Illinois. The column schema contains both inspection information and establishment information. Inspection-related columns include: *Inspection ID*, *Risk*, *Inspection Date*, *Inspection Type* and *Results*. Establishment details are recorded in columns: *DBA Name* (the legal name of the establishment), *AKA Name* (public alias or commonly known name), *License Number*, *Facility Type*, *Street Address*, *City*, *State*, and *Zip Code* of the facility.

Each row in the PPP dataset describes loan information, with columns detailing the loan amount, business type, lender names, race and ethnicity, gender, and location information of the company.

The Hospital and Flights datasets are sourced from previous work (Mahdavi and Abedjan, 2019). The Hospital dataset serves as a benchmark dataset in the data cleaning literature (Chu et al., 2013; Dallachiesa et al., 2013). It originates from the Hospital Compare website[7], maintained by the U.S. Department of Health and Human Services. The website provides data on the quality of care at over 4,000 Medicare-certified hospitals across the United States. The dataset includes key attributes such as provider number, hospital name, address, city, state, zip code, county name, phone number, hospital type, hospital ownership, and availability of emergency services.

The Flights dataset contains information about airline carriers, including details about their flights, scheduled departure and arrival times, and actual departure and arrival times. This dataset provides insights into flight schedules and potential delays by comparing planned and actual flight timings.

[7] http://hospitalcompare.hhs.gov

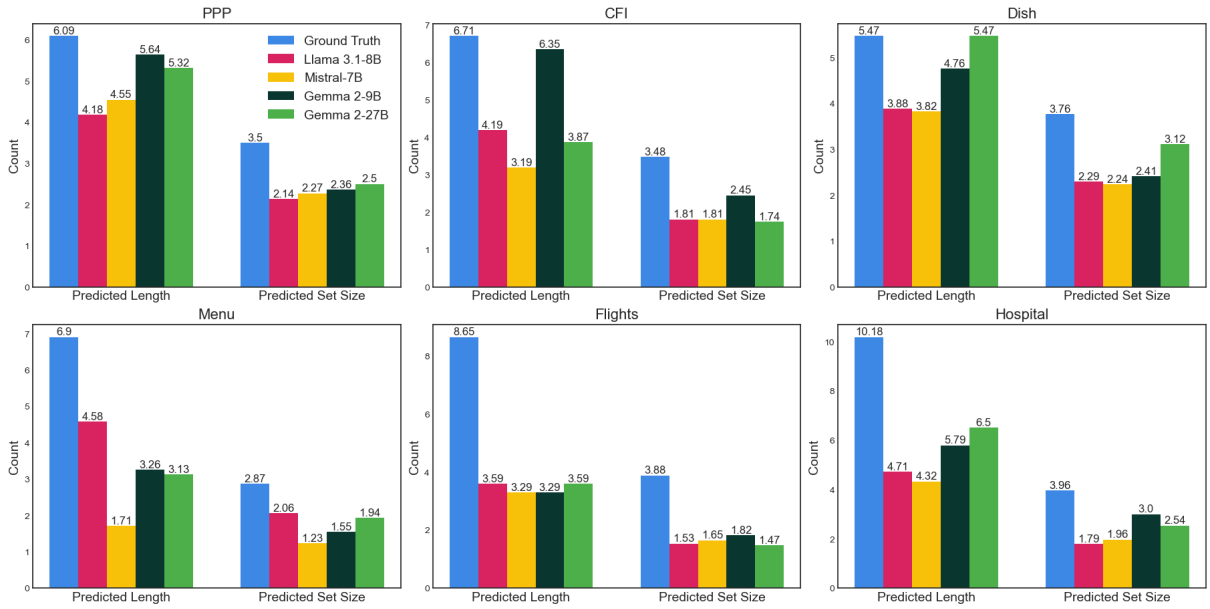

Figure 6: Workflow Length Across Six Datasets: Operation List Length refers to the total count of steps in each workflow, while Operation Set Length denotes the number of distinct operations within the workflow.

A.3 Mass_Edit & Regexr_transform: Arguments Introduction

For *mass_edit*, a list of dictionaries must be created, where each dictionary defines value replacements. In each dictionary, the *key-value* pairs represent groups of similar data instances that need to be standardized to a single, correctly formatted target value. Note that in OpenRefine, this operation is completed in multi-steps: (1). Data curators select clustering functions and parameters. (2). OpenRefine applies the clustering functions, groups similar values, and suggests a target value for standardization. (3). Data curators review the clustering results and confirm the target values. (4). The final value replacements are then applied. With LLMs, this operation can leverage the model's domain knowledge to recognize similar data instances, enabling the completion of *mass_edit* without manual verification.

On the other hand, the *regexr_transform* operation requires an "ad-hoc" Python-like code implementation as its argument. For example, the cell values in the target column *Year*—such as *Feyerabend,1975, Collins,1985*, and *Stanford,2006*–must be transformed to address the purpose of listing all published year information for the respective books. The example function (3) can be explained as follows: (1). "jython" signals that the following code is Python code. (2). The "return" statement is used to end the execution of the function. (3). The code between "jython:" and "return" parses and captures patterns in the cell values of the "Year" column, searching for four digits within these cell values.(4). The "value" parameter represents a single cell in the Year column.

```
jython: import re
match = re.search(r'\b\d{4}\b', value)
if match:
    return match.group(0)
```
(3)

A.4 Predicted Workflows Analysis

Figure 6 illustrates different workflow styles across models. Compared to ground truth workflows, predicted workflows are consistently shorter across all datasets. Similarly, LLM-generated operation sets are smaller than the ground truth. The shorter workflows reflect LLMs' ability to generate more concise and efficient operations for the same data quality issues. A smaller operation set size showcases the models' varying ability to learn and prioritize different operations, reflecting differences in how they optimize workflow generation.

A.5 Experiment setting

The model parameters for predicting the next operation are as follows: *temperature* is set to 0.1, *num_predict* is -1, *top_k* is 60, *top_p* is 0.95, and *mirostat* is set to 1. If the LLMs fail to choose the operation, the *temperature* will be adjusted to 0.3, while the other parameters remain unchanged. The parameters for arguments generation we use in *mass_edit* are: *temperature* is set to 0.2, with the same stop value.

A.6 Overall Performance Across Datasets

We summarize the detailed performance of each model across the six datasets in Table 8. The results highlight how well different LLMs perform under varying domain characteristics and data quality challenges.

A.7 Prompt Templates

Table 9: Generating Data Quality Report.

You are an expert in data cleaning theory and practices. You can recognize whether the current data (i.e., column and cell values) is clean enough to fulfill the objective. The evaluation process for deciding if the pipeline can end (Flag = True):
(1) Profile the column at both schema and instance level: Is the column name meaningful? What is the data distribution? Are values clearly represented?
(2) Assess four quality dimensions:
Accuracy: Are there obvious errors, inconsistencies, or biases?
Relevance: Is the column needed to address the purpose?
Completeness: Are there enough instances and minimal missing values?
Conciseness: Are spellings standardized? No different forms for same meaning?
(3) Return True/False for each dimension.
Only when all are True, return Flag as True; otherwise, return False and continue cleaning.
Example

LoanAmount	City	State	Zip
30333	*Honolulu*	*HI*	96814
149900	*Honolulu*	*HI*	
148100	*Honolulu*	*HI*	96814
334444		*IL*	
120	*Urbana*	*IL*	61802
100000	*Chicago*	*IL*	
1000.	*Champaign*	*IL*	61820

Purpose: Figure out how many cities are in the table.
Flag: True
Target column: City
Explanations: Accuracy: True (correct spellings for the same city names and same format in column City) Relevance: True (column City is relevant to the Purpose) Completeness: NA (with minor number of missing values in column City but it (1/7) can be ignored) Conciseness: True (incorrect variations do not exist in column City) Since there are no concerns (True or NA) with the quality dimensions, I will return True.
...

Table 10: Prompt introducing data cleaning operations.

You are an expert in data cleaning and are able to choose appropriate Operations to prepare the table in good format before addressing the Purpose. Note that the operation chosen should aim at making the data be in a better shape that can be used for the purpose instead of addressing the purpose directly. The Operations pool you can choose from contains *upper*, *trim*, *mass_edit*, *regexr_transform*, *numeric*, and *date*.
Available example demos to learn the data cleaning operations are as follows:
1. upper: The upper function is used to convert all cell values in a column that are strings into uppercase, fixing formatting errors for strings. ...For example,

id	neighbourhood	room type
46154	*Ohare*	*Entirehome/apt*
6715	*OHARE*	*Entirehome/apt*
228273	*ohare*	*Private room*

`Purpose: Return room types that are located near OHARE.`
`Target column: neighbourhood`
`Selected Operation: upper`
`Explanation: Improve conciseness: The format of cell values in column neighbourhood are inconsistent(mixed with different formats). Therefore, We use upper on column "neighbourhood" to make the format consistent as Uppercase.`
`Output: OHARE | OHARE | OHARE`
`...`

Table 11: Prompt Template: Column Selection Based on Purpose

Instruction: Select relevant column(s) from a table given its metadata and a natural language purpose.
Example:

```
{
  "table_caption": "List of
      largest airlines in Central
      America & the Caribbean",
  "columns": ["rank", "airline", "
      country", "fleet size", "
      remarks"],
  "table_column_priority": [
    ["rank", "1", "2", "3"],
    ["airline", "Caribbean
        Airlines", "LIAT", "Cubana
        de Aviaci\u00e3 n"],
    ["country", "Trinidad and
        Tobago", "Antigua and
        Barbuda", "Cuba"],
    ["fleet size", "22", "17",
        "14"],
    ["remarks", "Largest airline
        in the Caribbean", "Second
        largest airline", "
        Operational since 1929"]
  ],
}
```

Purpose: "How many countries are involved?"
Selected columns: "'['country']"'
Explanations: similar words link to columns : countries -> country.

Table 8: Overall Performance Results Across Datasets: {PPP, CFI, Dish, Menu, Hospital, Flights}. The highest results are **bolded**.

PPP

Model	Answer Dimension				Column Dimension	Workflow Dimension			
	Precision	Recall	F1	Similarity	Ratio	Exact Match	Precision	Recall	F1
Baseline (Raw Tables)	0.3318	0.3332	0.3309	0.5335	0.5962	-	-	-	-
Llama 3.1-8b (DP)	0.3334	0.3359	0.3329	0.5414	0.6272	0.2273	0.6364	0.3735	0.4364
Llama 3.1-8b (AutoDCWorkflow)	0.5812	0.5332	0.5449	0.7138	0.8178	0.2728	**0.9583**	0.6652	0.7559
Mistral-7b (DP)	0.3334	0.3359	0.3329	0.5401	0.6178	0	0.3182	0.1136	0.1626
Mistral-7b (AutoDCWorkflow)	0.3182	0.3225	0.3191	0.5387	0.6364	0.1818	0.8470	0.5864	0.6604
Gemma2-9b (DP)	0.3225	0.3287	0.3242	0.5420	0.6375	0.0909	0.3864	0.1826	0.2281
Gemma2-9b (AutoDCWorkflow)	0.4060	0.4133	0.4079	0.6247	0.7098	**0.3182**	0.9432	**0.6818**	**0.7661**
Gemma2-27b (DP)	0.3225	0.3287	0.3242	0.5420	0.6375	0.0455	0.7955	0.3197	0.4318
Gemma2-27b (AutoDCWorkflow)	**0.7373**	**0.7259**	**0.7273**	**0.8647**	**0.9038**	0.1818	0.8598	0.6735	0.7285

CFI

Model	Answer Dimension				Column Dimension	Workflow Dimension			
	Precision	Recall	F1	Similarity	Ratio	Exact Match	Precision	Recall	F1
Baseline (Raw Tables)	0.2511	0.2554	0.2318	0.4967	0.3829	-	-	-	-
Llama 3.1-8b (DP)	0.2357	0.2554	0.2248	0.4610	0.5295	0.0323	0.1935	0.0952	0.1184
Llama 3.1-8b (AutoDCWorkflow)	0.6444	0.6680	0.6484	0.7705	0.7238	0.0645	**0.9409**	0.5124	**0.6382**
Mistral-7b (DP)	0.2228	0.2366	0.2118	0.4875	0.4157	0	0.1129	0.0290	0.0436
Mistral-7b (AutoDCWorkflow)	0.3884	0.3441	0.3496	0.5720	0.5864	0.0323	0.7957	0.4161	0.5095
Gemma2-9b (DP)	0.2357	0.2554	0.2248	0.4610	0.5420	0	0.1935	0.0452	0.0727
Gemma2-9b (AutoDCWorkflow)	0.3568	0.3387	0.3272	0.5289	0.5743	0.0645	0.8081	**0.5452**	0.6178
Gemma2-27b (DP)	0.2679	0.2715	0.2463	0.4613	0.5528	0	0.3387	0.0935	0.1436
Gemma2-27b (AutoDCWorkflow)	**0.6911**	**0.7366**	**0.7026**	**0.7992**	**0.7455**	**0.0968**	0.9355	0.4844	0.6120

Dish

Model	Answer Dimension				Column Dimension	Workflow Dimension			
	Precision	Recall	F1	Similarity	Ratio	Exact Match	Precision	Recall	F1
Baseline (Raw Tables)	0.1471	0.1471	0.1471	0.6771	0.4863	-	-	-	-
Llama 3.1-8b (DP)	0.1471	0.1471	0.1471	0.6399	0.5505	0.0588	0.9118	0.2990	0.4294
Llama 3.1-8b (AutoDCWorkflow)	0.3269	0.2992	0.3078	0.6779	0.6684	**0.1176**	0.8676	0.5147	**0.6144**
Mistral-7b (DP)	0.2059	0.1711	0.1812	0.7036	0.5224	0	0.0588	0.0147	0.0235
Mistral-7b (AutoDCWorkflow)	0.2059	0.1711	0.1812	0.5482	0.5413	0.0588	0.7598	0.4314	0.5172
Gemma2-9b (DP)	0.2059	0.1711	0.1812	0.6664	0.5899	0.0	0.7941	0.2402	0.3608
Gemma2-9b (AutoDCWorkflow)	0.3235	0.2888	0.2989	0.6884	0.6066	0.0588	0.8088	0.5049	0.5889
Gemma2-27b (DP)	0.2059	0.1711	0.1812	0.6664	0.5899	0.0588	**1.0000**	0.3725	0.5196
Gemma2-27b (AutoDCWorkflow)	**0.6647**	**0.5977**	**0.6178**	**0.7774**	**0.7922**	0	0.6745	**0.5833**	0.5999

Menu

Model	Answer Dimension				Column Dimension	Workflow Dimension			
	Precision	Recall	F1	Similarity	Ratio	Exact Match	Precision	Recall	F1
Baseline (Raw Tables)	0.4172	0.4353	0.3851	0.4973	0.5104	-	-	-	-
Llama 3.1-8b (DP)	0.4672	0.4676	0.4282	0.5464	0.6124	0.0323	0.1935	0.1048	0.1301
Llama 3.1-8b (AutoDCWorkflow)	0.4688	0.4646	0.4408	0.5622	0.6542	**0.1935**	**0.8710**	**0.6231**	**0.7011**
Mistral-7b (DP)	0.4591	0.4622	0.4136	0.5165	0.5455	0	0	0	0
Mistral-7b (AutoDCWorkflow)	0.5000	0.4943	0.4632	0.5438	0.5984	0.0645	0.4892	0.2527	0.3154
Gemma2-9b (DP)	0.4661	0.4717	0.4325	0.5246	0.5675	0	0	0	0
Gemma2-9b (AutoDCWorkflow)	0.4685	0.4717	0.4345	0.5497	0.6666	0.0645	0.8226	0.4645	0.5661
Gemma2-27b (DP)	0.4661	0.4717	0.4325	0.5265	0.5675	0	0	0	0
Gemma2-27b (AutoDCWorkflow)	**0.6620**	**0.6213**	**0.6167**	**0.6849**	**0.7042**	0.1613	0.8172	0.5866	0.657

Hospital

Model	Answer Dimension				Column Dimension	Workflow Dimension			
	Precision	Recall	F1	Similarity	Ratio	Exact Match	Precision	Recall	F1
Baseline (Raw Tables)	0.1045	0.1142	0.1056	0.3498	0.4533	-	-	-	-
Llama 3.1-8b (DP)	0.2098	0.2120	0.2070	0.4466	0.6979	0	0.1786	0.0518	0.0793
Llama 3.1-8b (AutoDCWorkflow)	**0.5759**	0.5532	0.5622	0.7849	0.8911	0	**1.0000**	0.4583	0.6187
Mistral-7b (DP)	0.1045	0.1142	0.1056	0.3498	0.4533	0	0	0	0
Mistral-7b (AutoDCWorkflow)	0.2634	0.2656	0.2606	0.4750	0.6711	0	0.9226	0.4536	0.5914
Gemma2-9b (DP)	0.2098	0.2120	0.2070	0.4466	0.6890	0	0.1071	0.0387	0.0560
Gemma2-9b (AutoDCWorkflow)	0.2446	0.2375	0.2385	0.5084	0.7762	**0.1429**	0.9881	**0.7440**	**0.8392**
Gemma2-27b (DP)	0.2098	0.2120	0.2070	0.4466	0.6979	0	0.0595	0.0351	0.0440
Gemma2-27b (AutoDCWorkflow)	0.5750	**0.5723**	**0.5736**	**0.8258**	**0.9393**	0.0357	0.9524	0.6113	0.7323

Flights

Model	Answer Dimension				Column Dimension	Workflow Dimension			
	Precision	Recall	F1	Similarity	Ratio	Exact Match	Precision	Recall	F1
Baseline (Raw Tables)	0.0895	0.0557	0.0625	0.2668	0.0586	-	-	-	-
Llama 3.1-8b (DP)	0.4567	0.2823	0.3365	0.4111	0.4238	0	0.5000	0.1392	0.2157
Llama 3.1-8b (AutoDCWorkflow)	0.5487	**0.4517**	**0.4879**	0.5169	**0.7523**	0	**0.9706**	0.3833	0.5349
Mistral-7b (DP)	0.4947	0.3387	0.3876	0.4525	0.4269	0	0.3529	0.1029	0.1574
Mistral-7b (AutoDCWorkflow)	0.5812	0.4000	0.4593	0.4885	0.5794	0	0.8529	0.3578	0.4927
Gemma2-9b (DP)	**0.6635**	0.3894	0.4618	0.4779	0.6063	0	0.6471	0.1745	0.2728
Gemma2-9b (AutoDCWorkflow)	0.6047	0.3502	0.4253	0.4660	0.6208	0	0.9412	**0.4186**	**0.5542**
Gemma2-27b (DP)	**0.6635**	0.4090	0.4842	0.5081	0.6260	0	0.7941	0.2206	0.3395
Gemma2-27b (AutoDCWorkflow)	**0.6635**	0.4090	0.4842	**0.6586**	0.6194	0	0.9412	0.3431	0.4908