LLMs for Automated Data Cleaning: Improving Data Quality with AI Assistance

Pavithra Purushothaman
Matrikel Nummer: 1535949
fd0001571

June 28, 2025

1. Motivation

In almost every data engineering project I have worked on, there is one thing that takes more time than others: cleaning the data to make them useful. Data quality is an important part of meaningful analysis and decision-making. Inconsistent formats, missing values and duplicated records are common challenges in structured data such as CSV or Excel files. Before any analysis or model training, we have to fix missing values, remove duplicates and make sure everything is in the right format for usage. These are often time-consuming, repetitive and very manual processes.

I got interested in this topic because I wondered if LLMs could actually help make data cleaning easier? Instead of writing dozens of scripts by hand, maybe we could build a small tool where the model suggests what to fix or even applies some changes directly. It could save time and also help avoid any mistakes that may happen when everything is done manually.

However, LLMs also raise new concerns like hallucinations, introducing errors or misinterpreting tabular data. This project aims to assess if LLMs can be trusted to improve data cleaning quality and how they compare to conventional methods in terms of accuracy and reliability.

2. Research Questions

To guide this project, I came up with a few general questions:

- How can LLMs help find common data problems, like missing values or wrong formats in structured/tabular data?
- Can they do more than just find issues or they also suggest useful fixes or even apply them?
- How an interactive tool look like where the user stays in control but also gets help from the model?
- What are the limits? What can't these models do well when it comes to data cleaning?
- Can AI guided preprocessing outperform traditional rule-based methods?
- In which cases do LLMs hallucinate during data cleaning and how can these hallucinations be reduced?
- Can this tool improve data reliability?

3. Methodology

The project goal is to learn by building something simple and testing it out.

- **Dataset Selection**: Publicly available datasets with known issues (missing values, duplicates, wrong entries) will be used.
- **LLM Models**: Open-source models (e.g., `TableLLM`, `Gemma`) and interfaces via `Ollama` will be tested.
- **Prototype**: A tool using `Gradio` will allow users to upload datasets and receive cleaning suggestions powered by LLMs.
- **Comparison**: I'll also try traditional cleaning methods (like Python's pandas library), so I can compare results.
- **Observations**: I don't plan to do a big statistical study. Instead, I'll note what works, what doesn't and why. For example: Does the model find the right issues? Are its suggestions sometimes strange? Is the tool helpful overall?
- **Reflection**: At the end, I'll write about what I learned, what surprised me and what could be improved in the future.

4. Scope and Limitations

To keep things realistic, there are a few things this project won't do:

- The project work will not involve training new language models but will use and evaluate existing ones.
- The working prototype will be developed and deployed only for small to medium datasets, not on large or commercial databases.
- The research will focus only on structured data such as CSV and SQL based tables. Unstructured data (PDF, images, text) and semi-structured data (JSON, XML) are out of scope.
- The goal isn't to make a fully automatic cleaning system that works perfectly without humans. It's more of a research project to see what's possible and where it doesn't work.
- The project will focus only on data cleaning and preprocessing and will not include on data visualization or advanced analytics.

Deliverables:

- Research paper evaluating strengths and limits of LLM-based data cleaning.
- Prototype tool demonstrating LLM-powered preprocessing via Gradio interface.

5. Conclusion

Cleaning data is a necessary but often time-consuming step in most data engineering projects. I'm curious to see if large language models can make this part easier and faster. By building a small prototype and trying it out, I hope to get a better idea of where these models can help in practice and where they still fall short.