Beyond Data Cleaning

Enhancing OpenRefine with a self-hosted LLM

Hervé Letoqueux — CheckFirst
Dataharvest 2026 — Saturday, May 30 · 4:15–4:45pm
github.com/herve-checkfirst/DataHarvest2026-Refine_with_llm

Why OpenRefine in the first place

Reproducibility is the whole point

- Every action on a column is **recorded** in an exportable JSON operation history
- Re-run the same recipe on tomorrow's data — same result, every time
- A third party can **audit** your workflow step by step
- Self-documenting: the recipe *is* the methodology
- > *"If you cannot reproduce your column transformation, you cannot defend it."*
- For investigative journalism and regulatory work, an auditable trail is not a nice-to-have.

What GREL and clustering cannot do

Some columns need reading comprehension

- **Fact vs. opinion** — "a six-hour delay" is factual; "the flight was delayed" is not. A fuzzy, contextual boundary, not a pattern.
- **Aspect-based sentiment** — seat / crew / food / punctuality, each from the same paragraph of prose.
- > *"No combination of GREL, facets, or clustering can produce this output — it requires reading comprehension over hundreds of characters of unstructured prose."*
- Routes, free-text fields, contradictions between a rating and the words around it.
- This is where an LLM column earns its place — and **nowhere else**.

Why self-hosted, not a cloud API

Security & confidentiality by design

- **No data leaves your machine** — sensitive datasets, sources, leaks stay local
- **No API key, no per-token cost**, no rate limits, no vendor lock-in
- No third party logging your prompts or your data
- Works offline, on a plane, in a newsroom with no internet
- The stack: **OpenRefine + Ollama + a local model** (Ministral 3B), wired through the AI extension.
- Confidentiality stops being a policy you trust — it becomes a property of the setup.

Keeping reproducibility with a non-deterministic tool

The discipline that makes it defensible

- LLMs are non-deterministic by default — the same prompt can give different answers across runs.
- So, every transformation:
- **Pins the model** version
- **Fixes the seed** (42) and a **low temperature**
- **Documents** every parameter in the recipe
- Adds a **GREL guard column** to validate the output format and flag rows to re-run
- *> Pin the model, fix the seed, document the settings, write a guard column.*
- The audit trail survives the AI.

The limits of LLMs

Use as a filter, not a measurement

- **Error rate is real** — a 3B model: expect a 5–8% error rate on binary classification
- **Aspect bleed** — a complaint about food can wrongly mark crew as negative; worse on longer reviews
- **Understatement missed** — "the seat was adequate" read as positive when it is a soft negative
- **Format drift** — a few percent come back as `negative.`, `none mentioned`, off-schema
- **Built-in bias** — a tiebreaker rule can deliberately skew the distribution
- Mitigate: test on a small sample first, bigger model (7B ≈ half the error) for research-grade work, and **always** keep the human in the loop.

Takeaways

What to remember

- OpenRefine gives you **reproducible, auditable, self-documenting** data cleaning
- LLMs unlock transformations **GREL and clustering simply cannot do**
- **Self-hosting** keeps your data confidential and your costs at zero
- Discipline — pin, seed, document, guard — keeps it **defensible**
- LLMs are a **filter, not an oracle**: know the error rate, keep a human in the loop

Going further

Two ways to raise accuracy

- **Use a bigger general LLM** (e.g. 7B → 14B+, self-hosted)
- *Pros:* fewer errors out of the box, no training, swap the model and re-run, still local
- *Cons:* heavier hardware (RAM/VRAM), slower per row, still a generalist — caps out on niche tasks
- **Fine-tune a small model (SLM) for one task**
- *Pros:* high accuracy on *your* task, fast & cheap to run, tiny footprint, fully self-hosted
- *Cons:* needs a labelled dataset + training effort, one model per task, must re-tune when the task shifts
- Rule of thumb: **prototype with a bigger LLM, productionise with a fine-tuned SLM** when the task is stable and high-volume.

Questions & thank you

- **Hervé Letoqueux — CheckFirst**
- herve@checkfirst.network

- Repo, worksheets & dataset

  github.com/herve-checkfirst/DataHarvest2026-Refine_with_llm