Data minting: the systematic use of AI-readers for research data

Karl Rohe · University of Wisconsin–Madison · August 2026

Read the documents, fill in the form. This quiet chore underlies much of research. Historians mine archives; legal scholars classify opinions; systematic reviewers extract study details. Done by hand, the work is slow and error-prone: in one randomized comparison, extracting and verifying a single study averaged 107 minutes (172 with dual independent extraction),[1] and errors survive even careful processes.[2] Large language models can now do much of this reading: early evaluations report extraction accuracy comparable to human reviewers.[3]

These accuracy rates are promising, but limit the use at scale. For example, many researchers still feel the need to check every cell. Worse, the rate does not generalize to all studies: it was earned by one team's pipeline, and without a systematic process it promises nothing about new studies, new documents, new questions. And the finish line is hardly sufficient: matching humans means matching an error-prone baseline. Why limit ourselves to human abilities? The open question is not whether AI can fill the form; it is how to make the use of AI systematic enough to know which cells to trust.

A few years ago, everyone talked about "prompt engineering." The talk faded: chatbot providers optimized their models to guess what an imperfect prompt meant. That fix does not work for our research purposes. When a model extracts data, the written instructions are the measurement instrument: the extraction form, the screening criteria, the coding manual. I borrow a word from content analysis: the *codebook*—a list of questions, each with its definitions and examples. Whatever the codebook leaves ambiguous, the model silently imputes. A chatbot should do this; a measurement instrument should not. Readers guess differently, too: an instruction that can be read two ways will be read two ways, and both readings enter the data.

Data minting, the process I built at datamint.ing, is an attempt to make the use of AI systematic while remaining adaptable to projects with different needs, different documents, different topics. The minting is the systematic part: readers run, cells fill, the same machinery for every project. The craft is the adaptable part: piloting and testing the codebook until the mint performs well, different for every project.

Inside the mint, every question is answered by several readers, each a different AI model, each working alone with only the document and the codebook, each producing a trail of evidence, reasoning, and stated assumptions. An AI-analyst reviews the answers and evidence trails for each document. It marks, cell by cell, where the readers disagree.

Many people worry about AI hallucinations, the model inventing data. In my experience, hallucinations are very rare. When readers disagree, each side typically supplies real quotes and valid reasoning. What surfaces is the imputation itself, no longer silent—each reader resolved the same ambiguity their own way. As such, the core problem is lack of clarity in the codebook.[4]

Experienced teams already pilot their forms. With human coders, the loop is slow—train, extract, meet, adjudicate—so most teams can afford one round, and a disagreement resolved by discussion leaves no mark on the form unless someone writes it there. With AI-readers, a pilot round takes minutes. Each round lands on an inspection page, every disagreement opened to its quotes, its reasoning, and its diagnosis. The AI-analyst states in a single sentence what rule the codebook fails to specify—never which reader failed, never which document was sloppy. The AI-analyst proposes two different "notes" to append at the end of the question. By picking the note you prefer (or writing your own), you are answering a question about *your intent*; there is no right answer per se, only the answer specified by your research goals. These notes accumulate like case law documenting the edge cases.

This piloting sharpens what is being asked, not only how. And because the readers have no training materials outside the codebook, every resolution must be written into the codebook itself; nothing can live in anyone's memory. What remains is an instrument that any reader (human or AI) could follow, an artifact that you provide for others to build upon.

My conjecture, splits into two claims:

1. *Agreement licenses trust*: where readers agree, the extraction can be accepted without human verification.
2. *Disagreement directs attention*: where readers disagree, the cell needs a human decision and perhaps the codebook requires a clarifying note.

If both claims hold, verification effort concentrates where the ambiguity is.

If you keep an extraction form, screening criteria, or a coding manual, I would enjoy translating it into a codebook and piloting it with you.

The beliefs behind this work: datamint.ing/learn/highlights · karl.rohe@wisc.edu

[1] Li T, Saldanha IJ, Jap J, et al. *J Clin Epidemiol* 2019;115:77–89.

[2] Mathes T, Klaßen P, Pieper D. *BMC Med Res Methodol* 2017;17:152.

[3] Gartlehner G, et al. *Res Synth Methods* 2024;15:576–89, and *Ann Intern Med* 2025;178:1763–71.

[4] Documents are often unclear too, and it is fair to complain. But the codebook is the part we control, and it should specify what a reader does when a document is unclear. These are the edge cases.