Why we need an open format

Work in progress. This page is mirrored automatically from `rubicon/docs/open-format.md` in the Causal Map repository and changes as the argument does.

This should matter to you if:

- you are **any evaluation stakeholder** who is coming to believe that use of AI in evaluation is no longer a question of *how can we tell if it has been used* nor even of *should we* use it but of *how should we use it*
- you are an evaluation **commissioner** who wants to make sure your evaluator(s) use robust approaches for qualitative text processing, especially when using AI.
- you are an **evaluator** using or planning to use AI for qualitative processing of texts and you want to be able to show that your work is of high methodological quality, you avoided all the pitfalls and your work could, more or less, be reconstructed by an independent adjudicator. You'd like to be able to agree with your commissioner what workflows will be counted as robust. Or you'd like to be able to communicate to your stakeholders what you did and why, and how they can check it.
- you are an **academic or thought leader** interested in evaluation methods and how they are, and should be, applied.

Here we suggest an open format for describing qualitative text-processing workflows in evaluation. Further down we also describe software, Rubicon, which already implements it; but because the format is open, anyone could build software to do the same thing.

The problem

AI has arrived in evaluation for qualitative processing of texts, and it is staying.

Qualitative text processing is a key part of most evaluation: gathering and reading interviews and reports to answer questions which are often poorly defined. Some approaches (also) use quantitative methods, but even these are often embedded in, or have to combine with broader, qualitative work.

Until recently, the question has often been: are you using AI? Are you allowed to? Can evaluation commissioners tell if AI has been used?

Whereas now in late 2026, AI has changed the market and the profession. Just about anyone who wants to work competitively is using AI, and if they aren't now, they will be soon. And the spectre of hallucinated results and mitigating calls for a "human in the loop" has a reverse side: given the *potential* for AI when used well, why are you *not* using AI? A human checker might find errors in AI processing, but you can bet your bottom dollar on the reverse bet: that AI will find at least as many errors if asked to check *human* work.

Used well, AI raises quality on narrative material, more cheaply than anything before. It opens up the potential for as yet unimaginable new possibilities, some of them good. Used badly it produces junk — even faster. In the final evaluation report the two are impossible to tell apart: both sound confident and fluent, both quote passages that fit. What evaluation commissioners and evaluators should be asking is:

> What is the difference between a robust workflow and a junk one, and how can we ensure that evaluators will use / did use robust workflows?

What not to do with AI:

- Don't let an AI (or a human) make global judgements without clear procedures or criteria. Don't ask an AI to "summarise this document" or "what are the main themes here" or worst "is the project effective, according to the sources?"

So do not ask an AI (or a human) for a direct answer to a vague question. It will answer confidently, having decided for itself what you meant.

Vague questions are great! But…

Human progress is based on asking vague questions. But answering them robustly is all about *coming to an agreement about how to answer them*. Humans at least, and good AIs should push back on a direct question: "I can see where you are going with this, but as it stands, this question is not answerable. Let me help you break this question down so it is answerable; maybe you need to check back with your stakeholders, so this might take a while."

What to do: Break down a vague question into `well-defined` ones.

A well-defined question is one that two people with graduate-level generic social science skills would answer much the same way from the same material. The word is borrowed from the problem-solving literature, where a problem is ill-defined when the statement leaves parts of it open, so that no answer will be universally accepted (Reitman 1964), and where well and ill defined are the ends of a continuum rather than a binary choice (Simon 1973). We call a broken-down method for answering a question a **workflow**. It might be as simple as "count the instances of the word 'health' in each report and add them all up." But most interesting questions need more sophisticated workflows, perhaps splitting the task up into separate pieces, one for each subquestion, one for each source, perhaps iterating and looping and rejoining, and finally making an **evaluative judgement** about the value, significance or worth of the finding.

At Causal Map, we have long been proponents of a version of this "breaking down" strategy especially when using AI: see garden.causalmap.app. But that approach involves just one specific well-defined task: identifying causal claims. That fits perfectly into this kind of format. But it is of course not the only such task.

But wait, even humans don't yet agree how best to break down evaluation questions!

That's true. Many specific approaches like Outcome Harvesting or Process Tracing have manuals on how to apply them, but in practice most evaluators use, and need to use, varying kinds of bricolage to mix and adapt the methods to their needs (Aston & Apgar 2022; Apgar et al. 2024; Apgar & Aston 2025). But that does not mean that "anything goes".

So: an open format for qualitative text processing workflows in evaluation

Let's look at a way to specify workflows for breaking down one vague question into multiple, more clearly answerable ones.

Most evaluators are probably using Claude or ChatGPT right now, maybe in quite sophisticated ways they thought up themselves. Perhaps they keep track of decisions they make and the reasons for them, perhaps not. Who knows? Perhaps they used clever techniques to force their assistant to identify negative cases and address satisficing, perhaps not. Who knows? The reader of an ordinary evaluation report cannot see. Which passages were counted and which were passed over; what rule set the threshold at "more than half", how much agreement is needed to count as "many" or "substantially"; whether that rule was written before or after the first results came through; whether a second analyst would have even identified the same passages.

AI makes this problem worse, because a model reads a thousand pages overnight and nobody can check its working. But AI also makes the problem more easily fixable, because a machine that reads can be made to show every step.

This kind of format is agnostic as to whether humans or AI are following it. We call it **machine-readable** (like post-codes on envelopes) not because it has to be read by a machine, but because it is clear enough that even a machine could follow it, more or less.

There is no such format yet, and the nearest, REFI-QDA, standardises the coded result rather than the procedure that produced it. See this note: Is there already a standard?.

Nothing about it is exotic. It is a typed dataflow graph: steps with declared inputs and outputs, the order derived from those declarations, results that never change and know what they came from. Anybody who has met a build system or a data pipeline has seen this before. The format will be written as a JSON Schema, which validates the YAML and the JSON alike, so an implementer can point a validator at it and generate types in whatever language they work in. The schema is checkable in the same way: it declares which version of JSON Schema it is written in, the 2020-12 draft, and validates against that version's published meta-schema. So a reader can confirm our schema is well formed before trusting anything it says about a workflow, and no link in that chain rests on our say-so.

JSON and YAML, and not a shapes language. Three of the obligations are relations between parts of one document rather than shapes within it, and JSON Schema can express none of them: a quotation must appear at the character range it claims, every id a result was made from must name a result in the same document, and every citation must name a quotation in it. SHACL could state the referential ones declaratively, and RO-Crate already puts us within reach of it, but it validates RDF rather than JSON, the arithmetic and the character ranges would need SPARQL embedded in the shapes, and the reader we are writing for knows JSON Schema and does not know SHACL. The evidence that the plainer choice is enough is that two models from other vendors, handed nothing but a bundle and no specification at all, each wrote a working validator for the format in one go. So the format stays JSON and YAML, with a schema for the shapes and about a hundred lines of ordinary code for the three relations, and the conformance suite says which is which.

What this means is

- where evaluation workflows have not been agreed in advance, or even if stakeholders have no advanced knowledge of what workflows are being used, the evaluator can submit, alongside the report and the data, the actual workflows, expressed in an open format, with a link to the specification of the format. That means that even without resorting to dedicated software, a human (in principle) or an AI (easily) can at least approximately reconstruct the workflow and if desired even check it. An AI can be told: here is the data and workflow and schema, work out what it all means and how to reconstruct the workflow and answer my questions about it, criticise it, even re-run it .

Participation and reflection built-in

Participation (involving other stakeholders) and reflection (on the part of the analyst) are not add-ons to many evaluation workflows such as Outcome Harvesting; they are the most important part. But again, that does not mean "anything goes". An open format can set out a workflow with clearly marked phases which are *outside* the scope of any algorithm, where we just have to pause for participation and/or reflection to complete, before we can continue with a new result — or stop and start all over if that is the conclusion.

A workflow answers a single question; most evaluation tasks have many

Of course most evaluation tasks involve answering multiple, possibly inter-related, questions, not just one. In that case we just use multiple, possibly inter-related workflows. Here we will just focus on individual workflows.

High-stakes tasks often focus just one one question.

A workflow is like a flow diagram, but also a piece of text

[image not found: img/pasted-mtri0ofs-a55a.png]

A workflow can be expressed in different ways:

- as a flow diagram or "graph".
- in a special somewhat-human-readable text format called YAML
- less formally, in a human-friendly, easy-to-read format

In fact, the graph and the YAML are just different ways of saying the same thing.

What a workflow can express is set out in full further down.

AI assistance makes using such a format possible

Two years ago it would have been mad to suggest that evaluators or commissioners could ever be bothered to express an analysis plan (unless they had the sort of budgets available for RCTs) in terms of any kind of formal language. But an AI assistant can easily read and understand the format specification and help us to use it and explain what is going on - **but in a way we can check when we want to**. And once the format is specified, it's easy to build software to help evaluators use it.

A format has to be extensible

No-one is going to force evaluators or commissioners to a single set of tools. We can agree on the basic shape and extend it formally or informally. We can aim to provide a small, powerful, general set of text-processing tools which can be combined and recombined for different purposes, meaning that evaluators will most often not need to build new tools when they have a new idea, they can just build one out of existing tools, or most likely ask their AI assistant to do that.

Text judgements and rubrics

A step in a workflow might say to produce a text summary of a set of findings. Sometimes that is useful, even if it moves us away from being well defined; ask someone else and you'd get a different answer. Sometimes it helps to specify more clearly a simpler, less controversial kind of summary, like 'write a short version of the main points of each finding'.

Evaluators often use scales to condense a set of texts (and/or numbers) into a **single summary** (one out of a narrow range of options) according to rules expressed clearly in text; a **rubric** is a special kind of scale which attaches a different *value* to the levels of the scale, used for making evaluative judgements: is this level good, or good enough. Attaching value is an act which humans will certainly want to lead.

We do not suggest for a moment the application of pre-determined rubrics, like "`overwhelming` means at least 8 out of 10" or anything like that. The point is to give evaluators a tool for expressing such decisions and making them transparent.

[image not found: img/pasted-mtrig6ke-546d.png]

A level's description has to be detailed enough to apply. Ben Lawless and Julian King insist on this and we agree. Not being a bare name, under **Worth is declared by a person** below, is only the floor. The description has to carry enough detail and context that somebody can hold it against real material and say whether that material meets it. A description which only gestures at what it wants leaves the assessor reconstructing the reasoning case by case, which is the work a rubric exists to save. Our own consequence follows: several criteria will often bear on one judgement and have to be brought together, so the rubric says how they combine rather than leaving that to whoever reads the table. That combining rule is one of a great many things a workflow can express, set out under **What a workflow can express** below.

Sometimes you do not want criteria settled in advance at all. This is Julian King's point. It is a real case rather than a failure of nerve: the material can be richer than the question anticipated; until you have read it you may not know how the parts fit together or what the assembly should be. A format that supported only criteria fixed before the reading would be the poorer for it. So a workflow may stop at a table or a piece of prose without reaching a verdict, per **Build the thing, and not only the verdict** below.

(Pre)-registration as an option. Restricting freedom.

We asked above whether a rule that, say, set the threshold at "more than half" was written before or after the first results came through. Sometimes we have to, and want to, evolve our method and our criteria as we go along. But with high-stakes summative evaluations sometimes the commissoner might want to restrict the evaluator's freedom to tweak the method or change critieria.

So a format like this can optionally be **registered** with a commissioner, with a version, a date and a named person. What gets registered is the workflow, or as much of it as the commissioner asks for: which sources will be read and how they were drawn, how the text will be coded, what counts as "more than half", and the rubric that turns the counting into a verdict. The rubric is an obvious thing to register, because where "adequate" begins is nowhere in the evidence, and a standard written after the numbers arrive can be made to fit them. But a sampling frame settled after a first look, or a coding instruction narrowed until the finding firmed up, moves the answer just as far, and neither is any more visible in the report.

Registration is a dial rather than a gate. Register nothing and you still hand over a workflow somebody else can read — at inception, and/or later, and/or as it is iteratively developed. Register the sampling and the coding and you have committed to the descriptive half while leaving the verdict open. Register the rubric too and you have said what counts as good before you know what the material says. The commissioner asks for the setting the stakes deserve, and the freedom being given up is the freedom to change your mind without saying so. Everything else stays.

But doesn't that rule out iterating? No. In practice the question turns out not to fit the material, a distinction nobody expected forces new coding columns, a finding rests on one talkative respondent. Iterating means changing the method after seeing something, which is what registration exists to stop. The reconciliation is that **registration records ordering rather than commanding fixity**. Every version is dated and never rewritten, and every run records which version was in force, so a reader can see whether a threshold moved before or after the numbers arrived. That survives any amount of iteration.

So you can:

- choose to not register your workflow at all
- register your workflow but allow logged iterations

- register your workflow and not allow iterations

Software written to support this format should of course, when required, enforce versioning so everyone can see how the plan changed.

Agreeing rubrics before the evidence arrives is sometimes advised in the evaluation literature. Davidson advises drafting them before the evidence is gathered, as advice rather than as a dated record, and pre-registration proper is familiar to evaluators working with controlled trials and experiments. It has also been suggested for evaluation aimed at outcomes and analysis plans, which is the descriptive half (Nosek et al. 2018; Peck and Litwok 2024).

Extensions: Iteration and self-editing

It is not hard in principle to make this kind of format more powerful still, by allowing for loops, so that a workflow can iterate one or more steps a certain number of times or until a criterion is reached. Even more exciting - but arguably not so well-defined - would be to allow one step to edit the instruction of the following step in pre-defined ways, providing something more like an agentic workflow.

Avoiding bias

Iterating on what you have read is how the work gets better, and it is also how a method comes to fit one corpus and nothing else. You read the transcripts, the coding instruction gets narrowed until it finds what you saw, the threshold settles where the numbers happened to fall, and the report presents the result as though the method had been decided independently of it. Nobody has delibrately cheated, and the reader cannot see it either way.

Some mitigations.

Validation with synthetic material

A workflow can run perfectly and still be unable to tell a good case from a bad one. Nothing in its output shows that. We might be very reluctant to preregister or even tell anyone about our plan: what if it does not work?

So before the real material, it is now quite easy to use AI to generate synthetic documents in two sets, one formulated so that they should score well and one that should score badly, then run the initial workflow over both. If it separates the two sets as it should, you have some reason to trust it on the real corpus. If it does not, the problem is the method rather than the material, and you have found that out before it mattered.

Validation with synthetic material is not essential to this suggested format. It is just an example of how having tools to describe workflows accurately (and run them) opens up a whole vista of possibilities which were not possible just two years ago.

Hold material back.

- Develop the workflow on a subset of the material, or analogue material from a previous year or a similar program, then run the settled workflow over the rest.
- Where the workflow writes its own tests, split the corpus so that the half that wrote a test never scores it.

Test the instrument rather than the programme.

Run one instruction twice over the same ten sources and the flip rate is run-to-run variation; run two differently worded instructions over those ten and the difference is instruction sensitivity. Neither is a finding about the world, but both can be informative.

Being aware of absences

- **Search both sides equally.** (AI) coding workflows have to be very careful how to deal with absences and find ways to balance what was found with what was not found. Is seven mentions of depression across a fifty-page interview a lot or a little? Could we compare it with something else?

Anything that can legitimately find nothing must show that it at least looked. A check that passed because it matched nothing, a search pointed at the wrong place, a step that read half of what it was given: all three look exactly like a good result. So a workflow reports what it did as well as what it found, and the reader can tell an empty answer from a question nobody asked.

A negative finding is the hardest case. No quotation can evidence an absence, so demanding one pushes "nobody described a two-way benefit" into citing something irrelevant. A negative finding cites the search instead: what was looked for, where, and how much was read.

A proportion states its base, and the base includes the documents that gave nothing. Build a denominator from the passages already coded and a document that said nothing disappears, so "of the sources that gave me something" reads as "of the sources". On a sparse question, which unintended harm usually is, that inflates the headline figure where it matters most.

The absence an evaluator most often wants is the one no other method gives you: of the things the programme said would happen, which did nobody mention at all.

But we don't need an open format....

But surely the evaluator could simply add an appendix detailing the data and the prompts they used?

- No, for many reasons.

Unless the evaluator breaks down the work into "well-defined" questions, all we get is "someone asked a vague question and got a confident-seeming answer".

An appendix can be read. It cannot be run.

Readable is not the same as checkable. A dated rubric, a folder of prompts and an exported spreadsheet can all be read by a commissioner with the time and the inclination. None of them can be executed, so nothing in them can be checked against the documents they claim to rest on. Reading somebody's prompts tells you what they meant to do. It never tells you what the run did. The prompts happen in a context and unless you send the entire history, no cheating, it's hard to reconstruct what the model saw at what point.

The three things that go wrong are invisible in an appendix as well as in the report. A quotation that appears nowhere in the source. A proportion whose base does not include the documents that said nothing. A threshold settled after the numbers arrived. Each survives a careful reading of the method, because each is a property of the results rather than of the plan.

The evaluator can afford to be informal. The reader cannot. A commissioner receiving forty evaluations a year will not read forty folders of prompts, and neither will a synthesis team, a meta-evaluation or a sceptical colleague. They can run forty validators. Whatever is not machine-checkable at that scale is, in practice, not checked at all. The same applies to pre-registration: a version, a date and a named person need something to attach to, and a habit of working carefully cannot be registered.

Using a standard format helps mark out which parts of the workflow are locked down and which are free. Creativity and participation matter even more than they ever did. We can and should be able to be reflexive and participatory **and** rule-bound and auditable, at different points in our workflow.

What this does not claim. It does not make an evaluation good. On many jobs it would be overkill: one evaluator, a client who trusts them, modest stakes. They might still benefit from writing the workflow down at all.

Having an open format for AI use in text processing for evaluation is aimed at helping evaluators maintain confident, ethical human oversight over workflows which increasingly use AI, not because humans are better at checking but because in the end we vouch for our own work. It's about being persons of trust who maintain relationships with human stakeholders and earn that trust. In our profession, we need to maintain the role of the human evaluator as a last defence against a world in which machines just talk to machines.

References

Apgar, Bradburn, Rohrbach, Wingender, Cubillos Rodriguez, & Arias (2024). *Rethinking Rigour to Embrace Complexity in Peacebuilding Evaluation*. Evaluation, 13563890241232405. https://doi.org/10.1177/13563890241232405.

Apgar, & Aston (2025). *How Do We Define and Support Quality and Rigor in Causal Pathways Evaluation?*.

Aston, & Apgar (2022). *The Art and Craft of Bricolage in Evaluation, CDI Practice Paper 24*. https://doi.org/10.19088/IDS.2022.068.

Reitman (1964). *Heuristic Decision Procedures, Open Constraints, and the Structure of Ill-Defined Problems*. In *Human Judgments and Optimality*.

Simon (1973). *The Structure of Ill Structured Problems*. Artificial Intelligence, *4*, 181--201. https://doi.org/10.1016/0004-3702(73)90011-8.