Workflows gdoc

24 May 2025

https://docs.google.com/document/d/1nexgcqgtK-nTThCot7f35An1TRi3tdp4i4ZrHoPnfbc/edit?tab=t.0

**

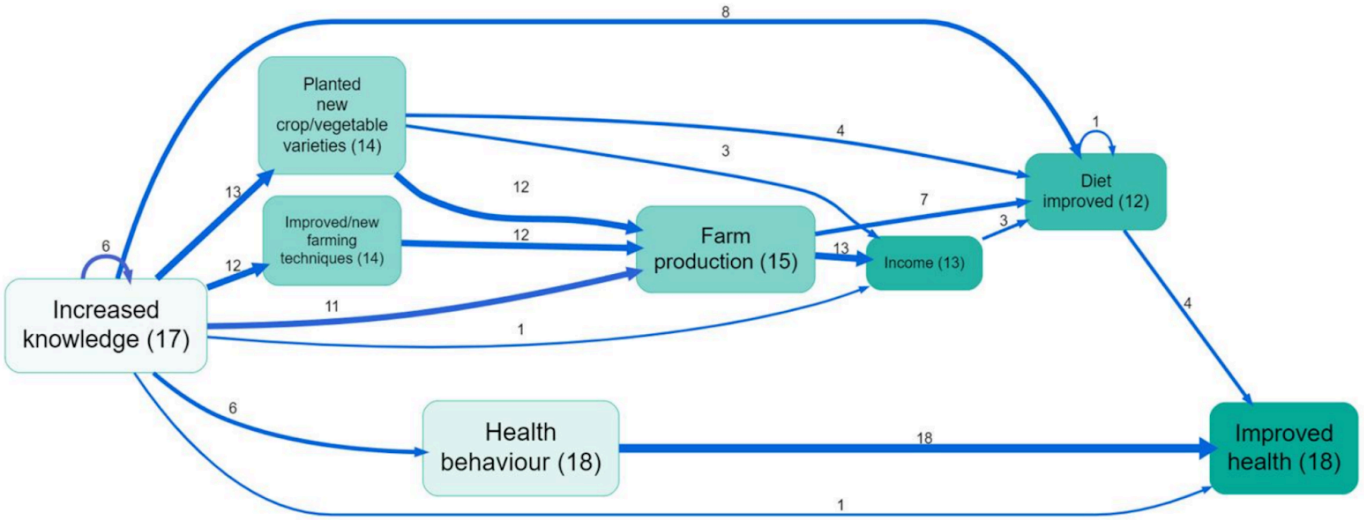
Making sense of mountains of text:

The Causal Map Workflows app.

What is Workflows?

Workflows is a tool used by Causal Map Ltd to make sense of large amounts of text data (interviews, reports) supported by generative AI.

At the moment, use of Workflows is not open for public use. Instead, we use it internally to provide verifiable and transparent analysis and reporting to clients. Clients can then access their workflows in the app to view and, if they wish, verify each step.



Which questions can Workflows help answer?

- Project impact through the eyes of stakeholders and other sources
- New, emerging and unexpected factors and outcomes
- Causal pathways

- OH: Synthesis of Outcome Harvesting processes
- MSC: Synthesis of stories collected for Most Significant Change processes
- QuIP: Synthesis of Qualitative Impact Protocol evaluations
- Contribution of projects to outcomes
- Stakeholders' mental models
- Comparisons between groups and time-points
- Vignettes: which stories are both remarkable and frequent?

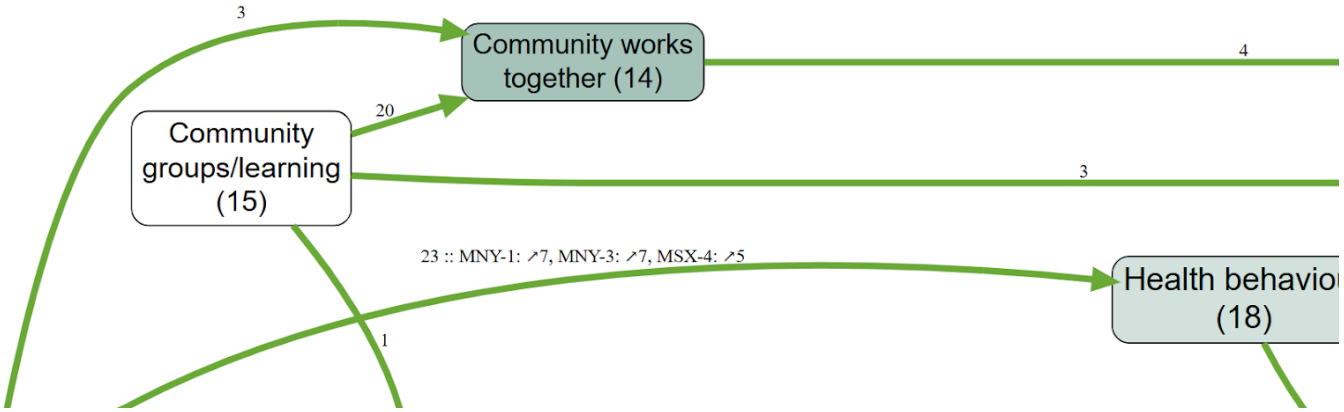
What do you get?

- Scalable Analysis: Process large amounts of text data quickly and efficiently.
- Transparent & Verifiable Results: Based on a clear, reproducible audit trail.
- Include more voices: Not just small samples of the easily accessible ones.

For Evaluators and Contractors

- Enhanced Bids: Offer state-of-the-art techniques to make your proposals more competitive.
- Time & Cost Savings: Reduce manual coding and analysis efforts.
- Reproducible Workflows: Guarantee that analysis steps are documented and repeatable.

Why Causal Map Ltd?



- Innovation in qualitative analysis and reporting
- Broad experience in evaluation and social science research.

Frequently Asked Questions

What does it cost?

We charge by day of our time, £695+VAT/day.

Most of the work isn't using the app. Most of our time is spent finding out what you want to know and what findings can help you, and iterating our reports until you are happy.

Caveats

Like humans, the AI processor makes mistakes and needs to be monitored. If you want an overall picture, the "noise" it can produce probably won't matter and you can get very good results almost out of the box. However at the moment, if you want to be sure that each individual piece of coding is correct, you'll probably need to tweak a not insignificant proportion of the coding it produces.

Why causal mapping?

At Causal Map Ltd., we promote causal mapping as a really useful and effective way of processing text which can be applied with surprisingly little adjustment to a great variety of evaluation tasks. Our Workflows software provides ready-made causal mapping steps, which can be used alone or seamlessly alongside other text processing tasks like thematic analysis. And the software can of course produce graphical causal maps. As the software can also be used without any causal mapping steps or visualisations, it could In principle be seen as also occupying a similar space to any of the market-leader CAQDAS applications which now offer AI assistance, but with more explicit, transparent workflows. However it is not our intention to compete directly with this kind of general-use software.

What is wrong with using AI as a "black box"?

At Causal Map Ltd, we use AI to both collect and analyse qualitative data. 🤖 We believe that the use of AI should also follow published guidelines to ensure transparent and valid results.

We are not comfortable with procedures which rely on giving the AI the freedom to make evaluative judgements. So we do not, for example, ask the AI "what are the main or most important causal stories in the document". We do not ask the AI to make summaries. Instead, we use the AI only as a tireless low-level assistant to exhaustively and transparently identify large numbers of individual causal claims (ideally, every single one of them) within the texts.

🔍 We follow established qualitative social science procedures for coding, with the aim that each step of the process from coding to analysis is transparent and verifiable. We don't rely on AI to simplify the coded data or decide, for example, which themes are most important.

AI-Powered automatic causal coding

The app is enabled with OpenAI's GPT-4o and 4o-mini and other models to automatically identify causal claims and connections within qualitative data like interview transcripts and documents.

- The AI follows detailed coding guidelines to exhaustively extract individual causal links, which are then verified against the original text.
- This AI-powered functionality allows for automated coding of qualitative data, significantly speeding up the process of identifying causal connections in texts.

The Story of Workflows

An app for making sense of text (interviews, reports ...) at scale. Using the power of genAI to synthesise and visualise the meaning of texts and answer high-level questions about them but with a verifiable data audit trail, step by step from text to final analysis.

A lot of evaluation work involves making sense of texts (interviews, reports...).

That's hard to do at any kind of scale, which often means using the most convenient and easy-to-reach sources and ignoring the rest. And it can involve weeks or months of work to create a coding framework and to code say 100 pages of text.

Generative AI seems to provide a solution to scaling.

That's a big challenge and a big opportunity. But there is a big problem to solve first.

To allow an AI to make evaluative judgements on its own is to rely on a big black box which is not transparent, or reproducible, or trustworthy, or verifiable. We may as well ask some random person we meet on the street to write our report for us.

The solution should have these three parts.

First: keep track using reproducible workflows.

Real-life research and evaluation assignments involve multiple tasks and sub-tasks. Traditionally, quantitative scientists have used workflows to keep track of their working in a reproducible way. No more "I found the graph but it needs updating and I can't find which version of the code produces it".

For example we can construct a documented workflow like a Jupytr workbook or an R-markdown document: a text document which includes the instructions for importing data, carrying out tasks and displaying output. It's like a human-readable computer program which when run produces outputs like charts and tables reproducibly.

Qualitative scientists too have been strong on documenting their research process, but as most of the steps involve human labour and human judgement, you can't just "rerun" a qualitative researcher's notebook and see the results reappear somewhere.

Now, AI is blurring the boundaries between qualitative and quantitative work. Some "qualitative" processes like thematic coding can now be carried out by machine, with some caveats. But, hacking around on the prompt until we get some sort of an answer and pasting the answer into a document somewhere – that's ok while we are learning and exploring, but it isn't reproducible or verifiable. An evaluation commissioner wouldn't accept quantitative work which we gave to a random person who claimed to be a statistician but who sent us some tables and graphs. And nor should they accept the results of "using an AI" if the working is not documented and verifiable.

Some evaluators are already exploring how to apply reproducible, workflow-based procedures to AI-supported work, mainly with Python.

However, just because an AI workflow is documented does not mean it is transparent. Which brings us to the second demand.

Second: breaking down big, vague tasks into small, easy ones so that we don't leave the AI to make big judgements all on its own

How can we reformulate hard tasks into easy ones, so we don't let the AI make significant evaluative judgements for us in an unverifiable way? Here's how.

- Break it down. Break down complex, poorly defined tasks (e.g. is this project effective?) into many smaller more clearly-defined tasks (e.g. does this paragraph mention behaviour change?) These tasks could be carried out by an army of trained human assistants, but we can use generative AI instead, and monitor its work.

- Build it up. Use agreed procedures to reconstruct higher-level evaluative judgements from the results of the simpler tasks using, and document the workflow transparently. Be clear about what parts of this reconstruction rely on human judgement.

The use of rubrics is a good example of this break-it-down-then-build-it-up strategy - usually rubrics are applied by humans, but when we have many cases, rubric assignment is a great task to hand off to AIs because we can easily monitor their performance and tweak the prompt and the rubric until the AI becomes as good as a human.

But how should I break down big tasks into smaller steps, and exactly which ones, and where and how should I employ an AI for the low-level steps, and when I am going to learn all that Python? This brings us to the third demand.

Third: providing a set of basic tools for the small, easy-to-automate steps

Workflow-based solutions are already being supported by AI service providers. Sort of. But if you ask, say, Google's NotebookLM to process a large table, it may or may not really produce results for each row.

Tools like this may produce plausible reports of their "thinking" which is a big step forward, but we don't really know exactly what steps were really carried out, only what it feels like telling us.

To do this properly, we need to learn to use genAI APIs and something like LangChain or Python and construct their own reproducible workflows using say a Jupytr notebook and selecting appropriate packages. But each kind of solution is different and can be time-consuming. It can be difficult to ensure or check that there are no black boxes left anywhere in the workflow.

So the third part of the solution we need is:

- A toolkit of the most common low-level tasks and recombination steps so that workflows for making sense of texts (including AI-powered steps such as coding, clustering similar texts and finding labels for clusters) can be presented as simply the application of tasks from this toolkit, one after the other.

Each step, whether AI-powered steps or ordinary data manipulation steps like sorting and filtering could also in principle be understood and followed by an army of human assistants. For example:

"Take these documents and break them up into paragraph-sized chunks. To each paragraph, add information about the age and role of the speaker. Ask the AI to decide whether each paragraph mentions changes in health behaviour. Reassemble all the examples into a single text, prepending each with the age and role of the speaker. Next, …"

All the work of handing off tasks to the AI, reconstructing the results, saving versions of the prompts and the workflow etc should not be carried out by a black-box AI but by ordinary computer software with published code. Only specific, low-level coding tasks should be given to the AI.

So at Causal Map Ltd we looked around for alternatives but in the end decided we had to build our own solution.

Our solution: workflows.causalmap.app

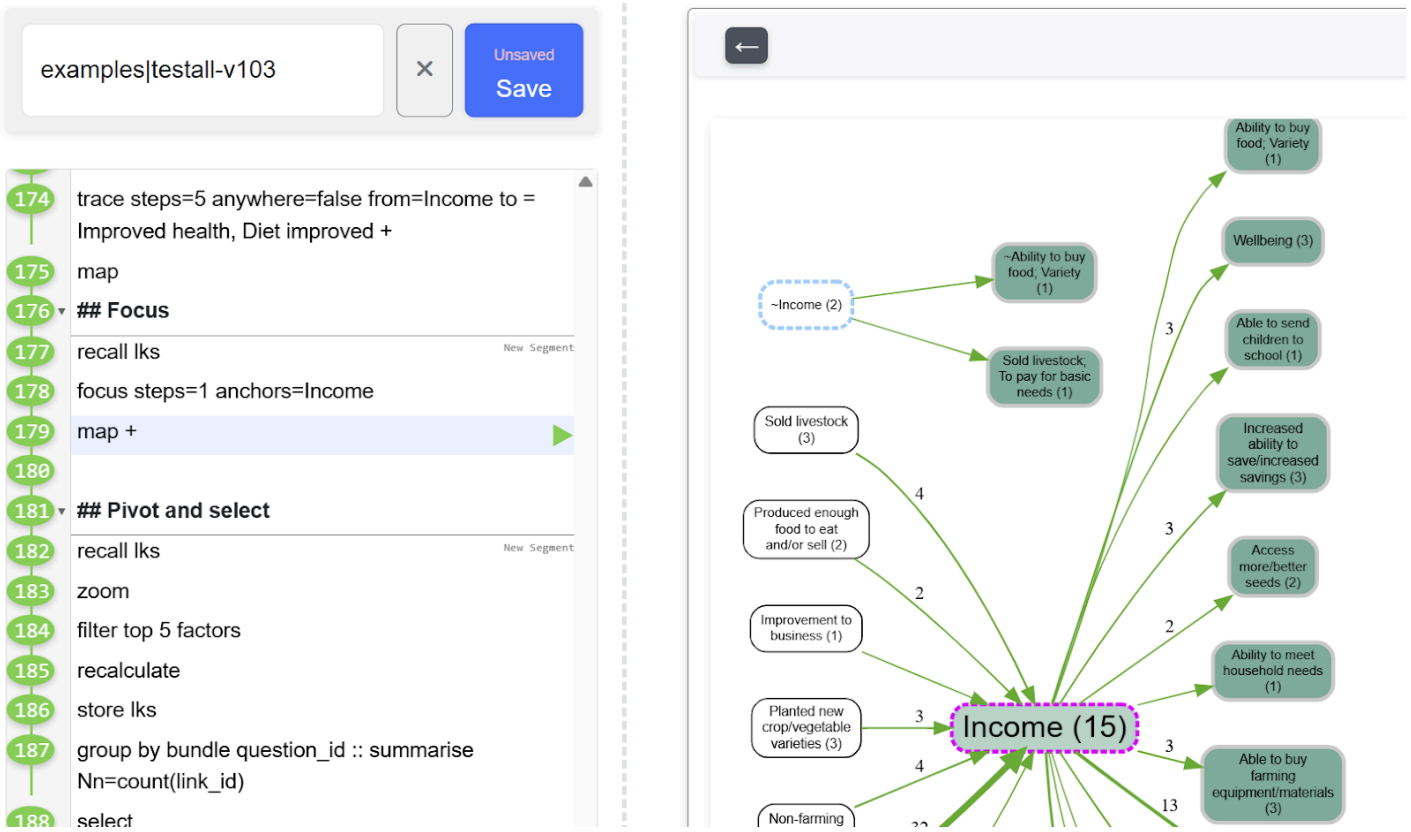
We present software (workflows.causalmap.app) which enables users to construct their own workflows for making sense of text, step by step from raw text to final outputs. Each workflow is simply a stored piece of text: a list of steps, one per line, described in a basic scripting language which is more or less human-readable.

Each line in the command editor has a data table associated with it.

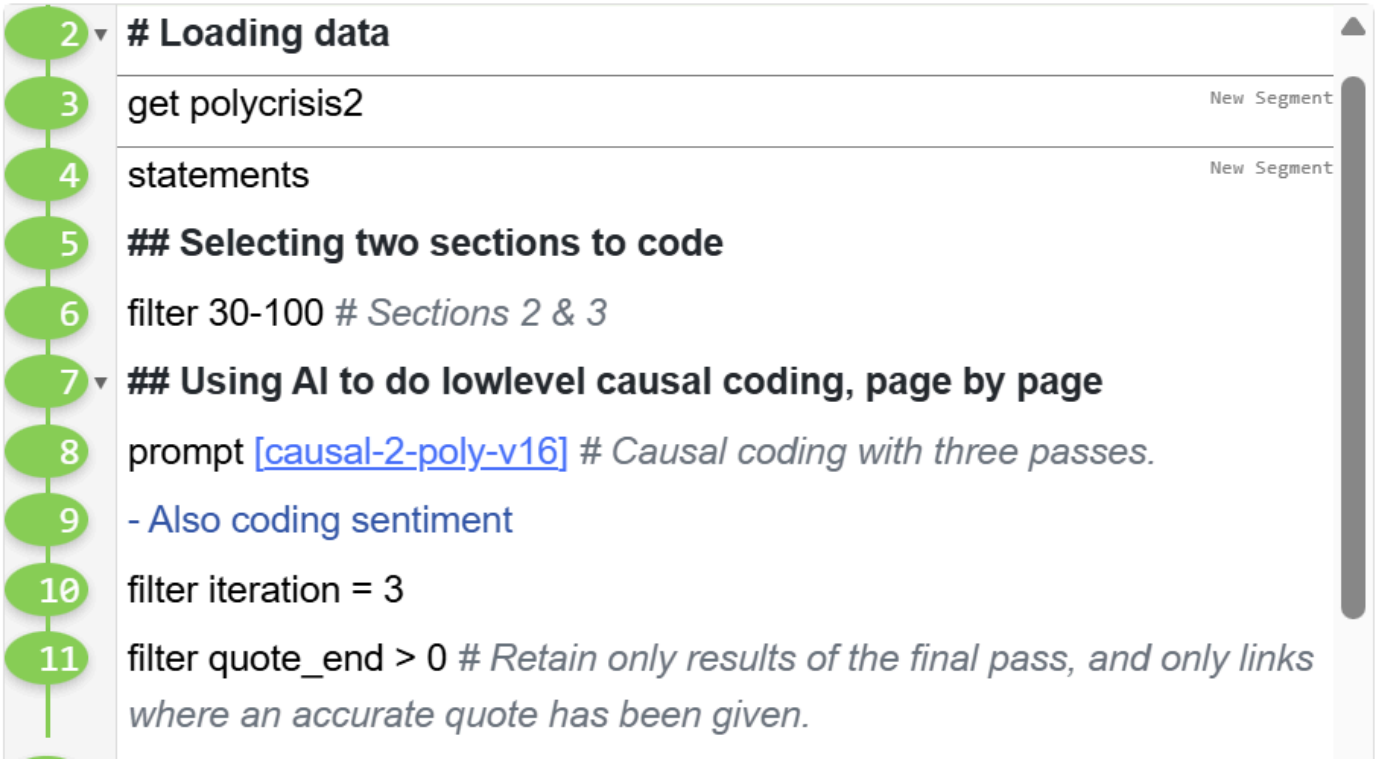
The workflow starts by importing data from storage, from a web page or a Qualia interview, or from a previously uploaded set of documents, resulting in a table of data.

Each subsequent line operates on the current data table and applies a verb (pivot, filter, sort, code ...) to that table, resulting in another table (or an output like a chart or a map).

The user can click on any line to see the result of the workflow up to that point, usually the data table itself or alternatively a visualisation of it, like a causal map.



We can insert AI operations into the workflow at any point. We can save and re-use prompts which work line by line on the current data table. Each prompt returns one or more rows with whichever columns we ask for, which are reassembled into a new table. Using AI prompts becomes just like any other manipulation of a data table.



Features

Currently, the following features are implemented:

Data input:

- Loading text from a database or a Qualia interview or directly by scraping a URL
- Uploading one or more PDF or Excel (xlsx) documents

Steps useful for AI processing:

- Chunking and unchunking text eg combining shorter texts into longer ones
- Managing and editing different AI prompts and sequences of prompts, processing them in parallel, and caching the results in a noSQL database. The user does not need to explicitly “save” different analyses: if it’s been done before, the app will find the results and present them without need for reprocessing.
- Repeating multiple steps in a loop
- Clustering individual texts into groups with a similar meaning and finding labels for them
- "Magnetising" texts: given a set of text labels designated as “magnets”, relabelling each of a large set of individual texts with its closest magnet, if any are close.

Causal mapping steps: All the same functionality as in the Causal Map app, e.g.

- Filtering links in a causal map, for example to show the most frequently mentioned factors or links,
- Zooming in or out when using hierarchical coding
- Tracing paths and threads from one set of factors to another
- Focusing on a particular set of factors
- Relabelling factors e.g. with cluster labels
- Calculating and displaying differences between groups

Common steps for manipulating tables:

- Selecting and renaming columns, filtering by value or meaning, sorting, performing simple calculations, combining columns, appending, merging and pivoting tables

Output steps:

- Printing tables in text format, using templates if required
- Providing tables which can be manually filtered, sorted and exported
- Displaying causal maps
- Storing individual results for use later in the workflow.

Evaluative judgements and evaluator responsibility

A clarification: We are not saying that breaking down evaluation tasks into this kind of workflow frees evaluation from human judgement. Quite the opposite. We think the discipline of being more explicit about how evaluative judgements are made is a good thing, whether the steps are carried out by humans (as in the past) or with AI assistance. If you didn't have a clear workflow from data to judgements before AI, DON'T lean on the black box of the AI to cover that up. Instead, make the workflow explicit / human-verifiable and then use the AI to speed up the process.

Even where a workflow does not include explicit human input at any stage, the decision to use this particular workflow, the details of its design, quality assurance checks, and, crucially, interpreting the results so they will be correctly interpreted by the evaluation audience - these are all the responsibility of the evaluator.

Current status of the app

- We don't have immediate plans to open up the app for general use.
- The backend - the Python code which implements the steps asynchronously to deal with long tasks.

- Documentation is still work in progress, the causal mapping steps are already covered in the Causal Map Guide.
- The frontend UI provides:
- A powerpoint-like presentation mode so that external viewers can step through a workflow and view intermediate and final results, together with comments, section headings etc.
- Ability to save and recall different workflows and individual custom texts e.g. prompts, lists of magnets, with good versioning, to address the problem of "how did we get that particular map?"
- A searchable gallery of workflows
- Full-screen mode
- Authentication via Google Firebase and Authorisation based on Causal Map permissions to access the files stored there
- Coming soon: an AI assistant to write the steps for you.

Technical details

Backend: Python, Quart (asynchronous server for time-consuming processes), hosted at Heroku. NoSQL caching at MongoDB. SQL data stored at Heroku. Access to Causal Map SQL data. OpenAI API with various models available. Implicit generation of embeddings using OpenAI's text-embedding-3-small, cached at MongoDB.

Frontend: Alpine, Axios, CodeMirror editor, Tabulator tables ...

Why a new app?

Causal Map 3 is written in R-Shiny which is not a platform of choice for AI. CM3 is a user-focused platform with lots of buttons and sliders and a fixed approach to applying filters.

Whereas Workflows:

- Uses Python which is where all the new things are being created. It is much easier to scale and maintain
- It is easy to write new functions without having to provide additional UI (corresponding buttons, sliders, etc).
- Functions aka commands can be arbitrarily chained
- We can experiment and optimize new approaches without having to rewrite the UI every time

- Common patterns of commands can be easily combined into templates
- Prompts and workflows have version numbers
- There is no need to save different versions of coding results. Thanks to noSQL caching, if a particular prompt has been run before, the app will find the cached results.
- No restriction on multiple users working with the same data at the same time
- AI tasks and other tasks can be arbitrarily mixed and chained.
- Applying an AI step should feel no different from adding any other filter in Causal Map

How we do automated causal mapping

We import the texts to be coded, together where appropriate with meta-data (such as age and gender of respondent etc) into Causal Map Workflows.

We agree on an approach for the coding e.g.

- Purely “zero-shot” coding which allows the app to develop its own generic factor labels (“empirical codebook”).
- This can be augmented by a “global orientation” for example telling the AI to pay particular attention to power relations between people, and code them in the light of a particular theory of power relations.
- We may need to provide background information to the text. Or say that a particular organisation might be referred to in different ways, and to code all mentions in the same way.
- Or we might say that we are particularly interested in chains in which some intervention (someone or some organisation intervening or trying to provide help in some way) is at (or near) the beginning of the chain, and in which important outcomes (like people's lives getting better or worse) are at (or near) the end of the chain.
- We will usually test an initial set of instructions on a subset of the text and possibly then revise them.
- After each actual wave of coding, we conduct auto-clustering. Your tests reach the AI broken up into many separate batches. When employing open coding, with zero-shot, the factor labels will probably form a pile of overlapping concepts, with similar ideas expressed in different ways. So we use auto-clustering to form the factors into clusters of very similar ideas, and find a shared label for all of them. This process of auto-clustering is very useful to get a sorted overview of the kind of material the AI is identifying. When applying automatic clustering, we take care to set the “height” aka “strictness” parameter to balance the desire to have only a small number of large, simple factors with

the caveat that the clusters should not contain any pairs of factors with importantly different meaning.

- We will usually also, after each wave of coding, produce some simple causal maps to give an overview of what the AI is finding with the current instructions.

- Zero-shot coding can be followed by a second phase in which the list of factors from the empirical codebook, probably auto-clustered as described above, and possibly adjusted by the client, are given as suggestions in a new semi-closed set of instructions. The coding is re-run with this new codebook. This ensures more thorough and consistent coding of the factors of particular interest. We can tweak the instruction to insist more or less strongly that only the suggested codes should be used.

- Alternatively you can skip zero-shot coding and start with your own codebook. This can be simpler and is more suited to more focused, closed research questions; but it does also hinder identification of unexpected concepts.

- We recommend that your codebook is a simple list of relevant causal factors. It is also possible to provide more detailed examples of the target factors, but this can hinder identification of unexpected details.

- In theory it is even possible to start by getting an expert to hand-code some of the texts and then producing a customised instruction based on this coding as an example. However in practice this can be complicated and is only really suitable if you are looking for some very specific information (and in this case it might be easier for you to use a tool like Ailyze instead).

- At each step we will keep an eye on the quality of the coding:

- Recall (is the AI finding a sufficient proportion of all the causal claims in the text?)

- Precision (is the proportion of inaccurate AI coding sufficiently low?)

- Is there any evidence of any kind of bias (the AI is consistently missing some codings of a certain type, or is consistently miscoding in a particular way). Otherwise we can assume that even if the recall and precision are not very high, the overview causal maps will be accurate in aggregate.

-

- If you need a high degree of accuracy of causal coding (identification of causal claims), it is possible to manually check and if necessary correct each coding. We can help you to do this yourself. This can take a long time, from many hours to many days.

Assuming we were successful in helping you turn your research aims into an appropriate research design, producing the results should now be quite straightforward, producing:

- overview maps to answer research questions (see above)

- corresponding quotes to illustrate the maps
- corresponding tables of frequencies

We can produce additional and more detailed outputs for you, but this will mean additional time.

**