🌻 Task 2 – Causal coding – minimalist style.

Chapter Contents.

> Standing on the shoulders of giants

In this chapter we present some of key general principles about how to do causal mapping which we at Causal Map Ltd (and, most of the time, at BathSDR) have adopted.

This is a very restricted yet powerful **minimalist** approach which we have also called "barefoot" or "naïve" coding.

In the next chapter Tasks 2 & 3 -- Introduction we look at specific conventions to make causal coding simple and powerful.

PAGES IN THIS CHAPTER

Minimalist coding does not help with blockers and enablers

> Thanks for inviting me to the discussion, James. **I'll start by describing the "minimalist" approach** to coding causal statements used for QuIP and developed originally by James, Fiona and colleagues at BathSDR and developed and further formalised at Causal Map Ltd in collaboration with BathSDR. This formalisation lives inside the Causal Map app. Then I will try to **answer the question of whether it can help us deal with more complicated constructions like enabling and blocking** and whether this could help us with mid-range theory. As an appendix I'll add a more detailed overview of minimalist causal coding.

Our approach is minimalist – we code only bare causation

> Why we stick to bare causation in causal mapping.

Our approach clearly distinguishes evidence from facts and does not automatically warrant causal inferences

Our approach is minimalist – factors are not variables

> Many or most causal mapping approaches, including Causal Loop Diagrams, also code the perceived strength of a causal link. This means that the factors become variables which can take values between, say, low and high or positive and negative, and we can make a much broader range of inferences using some form of numerical modelling. This can be seen as the extreme reproducible end of our spectrum and borders on quantitative approaches.

1a A minimalist approach to coding helps capture what people actually say

1b A minimalist approach to coding makes aggregation easier

> We just argued that [[0130.1a A minimalist approach to coding helps capture what people actually say]]. But even if you did succeed in imposing some special logical features on your data -- for example, coding necessity and sufficiency -- you'd probably find that most of your data didn't fit well with these special features. When it comes to aggregating medium or large amounts of coding, you wouldn't find it very useful.

1c A minimalist approach to coding does not code absences

> One thing which makes causal mapping a fundamentally qualitative approach is that we do not code absences.

Our approach is minimalist – we do not code the strength of a link

In a causal mapping dataset there is no need for a special table of factors

Factor labels – a creative challenge

> Where do the labels for the causal factors come from? As with ordinary QDA and thematic analysis (Braun and Clarke, 2006), approaches vary in the extent to which they are purely exploratory or seek to confirm prior theory (Copestake, 2014). Exploratory coding entails trying to identify different causal claims embedded in what people say, creating factor labels inductively and iteratively from the narrative data. Different respondents will not, of course, always use precisely the same phrases, and it is a creative challenge to create and curate this list of causal factors. For example, if Alice says ‘Feeling good about the future is one thing that increases your wellbeing’, is this element ‘Feeling good about the future’ the same as ‘Being confident about tomorrow’ which Bob mentioned earlier? Should we encode them both as the same thing, and if so, what shall we call it? We might choose ‘Positive view of future’, but how well does this cover both cases? Laukkanen (1994) discusses strategies for finding common vocabularies. As in ordinary QDA, analysts will usually find

> themselves generating an ever-growing list of factors and will need to continually consider how to consolidate it – sometimes using strategies such as hierarchical coding or ‘nesting’ factors (as discussed in the following section).

Factor label tags – coding factor metadata within its label

> For example you might want to code the respondent's happiness at work as different from yet similar to their happiness at home. With a factor table, you could have a field called `label` = "Happiness" and another, say `context`, which is = either "Home" or "Work". This is what we do with the links table in Causal Map, where we do have some hard-coded (but optional) fields and some user-definable fields.

Factor labels – semi-quantitative formulations can help

> It might be tempting to try to formulate all factor labels in a strictly similar way, using for example language like increased probability of … or positive change in … in every case. But it is difficult to identify and agree on a satisfactory template for doing this which will capture enough of the way people really make causal explanations (in the way that quantitative social scientists hope to measure everything just with continuous variables). This is always a balancing act, but we encourage you when in doubt to stick fairly close to the actual language your sources use (so-called “in-vivo” coding), and don’t be *too* worried if your factor labels are different from one another grammatically (e.g. some express a difference like improvement in X and some do not).

Factor labels – do not over-generalise

> When you are creating factor labels for re-use across different causal claims, you should usually take care to keep them specific: make them no more general than they need to be.

Coding with and using link metadata

> In our implementation of causal mapping in the Causal Map app, [[0130.2 Our approach is minimalist -- we do not code the strength of a link]].

Link metadata – Sentiment

> What is it for?

Link metadata – Time reference

> It is often useful to code a time reference. We often conflate time with hypothetical status, e.g.
>
> - hypothetical past/present
> - factual-past/present
> - future-planned

- future-hypothetical

Link metadata – quality of evidence

Research on LLMs' ability to detect causal claims

> The capacity of Large Language Models (LLMs) to identify, generate, and reason about causal relationships in ordinary language represents one of the most significant, yet enigmatic, developments in artificial intelligence over the last decade. As noted by observers of models since the release of ChatGPT (based on GPT-3.5) and its successors, these systems exhibit a "native" ability to process prompts involving influence, consequence, and mechanism without requiring the extensive few-shot examples or rigid schema engineering that characterized previous generations of Natural Language Processing (NLP). This report investigates the trajectory of this capability from 2015 to 2025, deconstructing whether this proficiency is a serendipitous artifact of scale or the result of specific, albeit implicit, training choices.