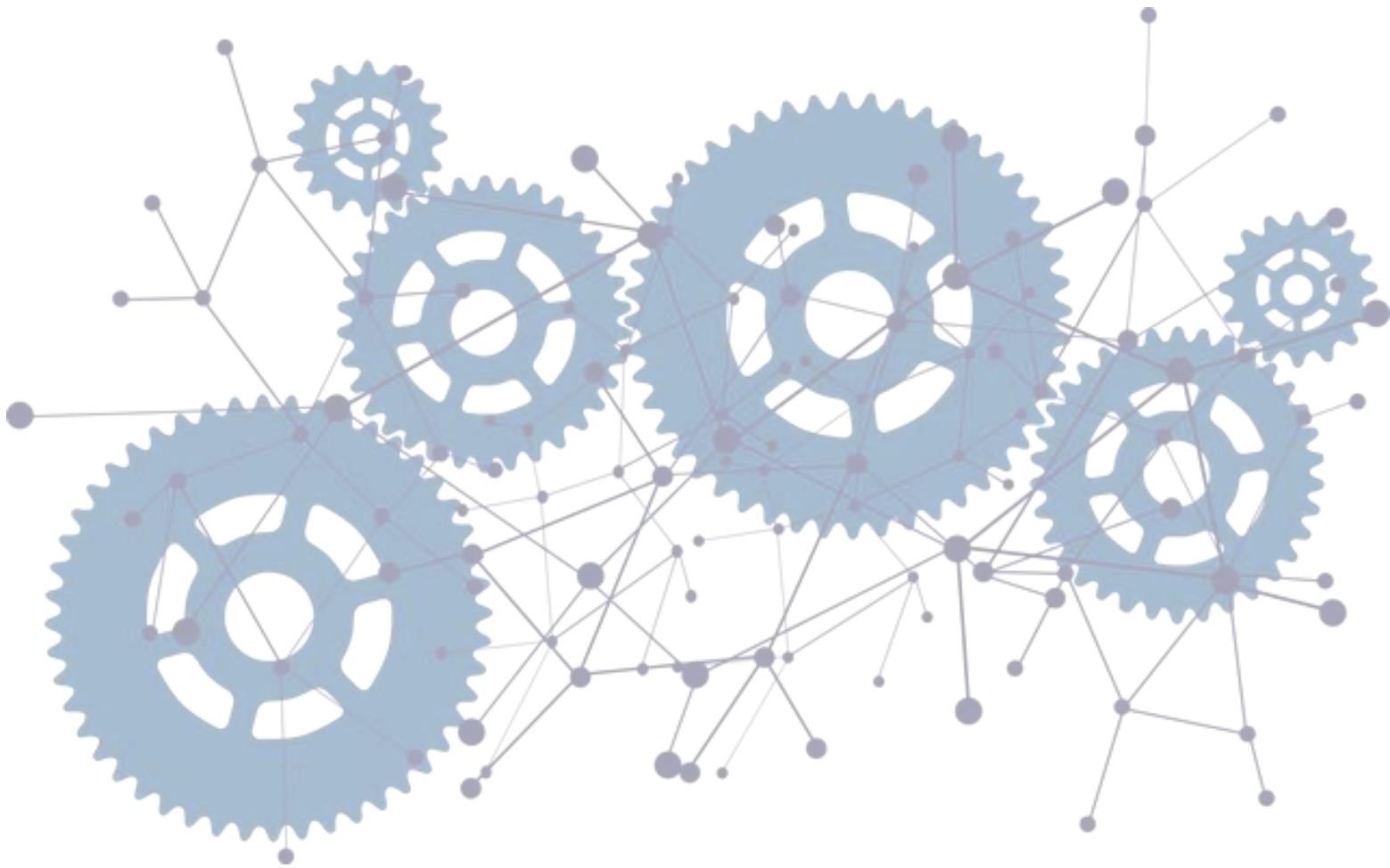


Guidance on using large language models to extract cause-and-effect pairs from texts for systems mapping

Jordan White¹ and Pete Barbrook-Johnson^{1,2}

¹ Institute for New Economic Thinking, University of Oxford

² Centre for the Evaluation of Complexity Across the Nexus, University of Surrey



Authors

Jordan White, Institute for New Economic Thinking, University of Oxford

Pete Barbrook-Johnson, Centre for the Evaluation of Complexity Across the Nexus,
University of Surrey

Contact

We welcome comment on this report. Please email jordan.white@bristol.ac.uk or peter.barbrook-johnson@ouce.ox.ac.uk.

Please cite this document as:

White, J., & Barbrook-Johnson, P. (2025). *Guidance on using large language models to extract cause-and-effect pairs from texts for systems mapping*. Available at www.cecan.ac.uk/resources.

This work is licensed under the Creative Commons Attribution 4.0 International License



To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

You are free to:

- Share - copy and redistribute the material in any medium or format
- Adapt - remix, transform, and build upon the material for any purpose, even commercially. This license is acceptable for Free Cultural Works. The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

- Attribution - You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- No additional restrictions - You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Executive Summary

This guidance explores how researchers and analysts can use large language models (LLMs) to support systems mapping exercises.

Systems mapping is a suite of methods used to develop visual causal descriptions of how systems work, which can help analysts and policy makers understand the implications of change or intervention in the system. They are widely used by policy analysts to inform policy design and evaluation.

Systems mapping can be a labour-intensive process requiring manual reading of policy and research documents or running mapping workshops with stakeholders. Using LLMs to supplement a systems mapping exercise can potentially help analysts incorporate a broader range of materials, leading to more comprehensive and accurate representations of policy systems. They also potentially speed up the mapping process.

This guidance document presents some of our thinking and experimentation on the role of LLMs in systems mapping. We begin by introducing LLMs, explaining how they work, describing how to access them, and highlighting associated risks. We then provide step-by-step guidance explaining how to use a popular LLM to extract the cause-and-effect pairs, which form the network data needed to produce a preliminary system map. Importantly, we discuss ‘prompting’ techniques that appear to improve the output accuracy of LLMs.

Disclaimer

The LLM products discussed within this guidance were the most popular models at the time of writing, however many other free-to-use and subscription products are available. As authors, we bear no affiliation with any of the developers of any products mentioned and more success may be seen by using our method with other products.

Table of Contents

Introduction.....	5
Which LLM should I use?	8
Basics of how to use an LLM	11
How does an LLM work?	12
Are there any risks associated with LLMs?.....	14
Guidance: How to use LLMs to extract cause-and-effect pairs from texts	18
Using an LLM as part of the systems mapping process	18
1) Prepare the policy document	20
Converting a document into plain-text format.....	20
Work with one document at a time	21
2) Provide the LLM with a prompt and a policy document.	22
Example prompt	31
Unsuccessful techniques.....	33
Submitting your prompt and document to the LLM	36
3) The LLM processes your prompt.	37
4) The LLM provides a response.....	37
Troubleshooting	37
5) Export the list of cause-and-effect pairs.	39
6) Creating a system map using PRSM	42
Example system map.....	42
Importing the list into PRSM	46
Method and Limitations	47
Authors.....	52
Contact information	52
Resources	53
References	54
Appendices	56
Appendix A: Ranking of highest quality LLMs as of December 2024.	56
Appendix B: Generating system map visualisations natively within ChatGPT.	57

Introduction

This guidance focuses on using artificial intelligence (AI) in the form of large language models (LLMs) to automate the extraction of pairs of causes and effects from pieces of text derived from policy documents or research reports and papers.

The motivation behind doing this is to support policy analysts and researchers in quickly producing preliminary maps of policy systems based on relevant documents which might contain causal language about that system. Currently, building preliminary system maps can only be done in relatively labour-intensive ways, either through manually reading such documents, or running mapping workshops with relevant stakeholders.

We use the word ‘preliminary’ here very intentionally. The outputs of LLMs in the context of systems mapping should only be used as a first estimation, as they are likely to have errors and omissions. Our expectation is that these preliminary system maps be used to seed further exercises, rather than be used as final outputs themselves.

In this guidance document, we focus on extraction of causal statements, and do not provide extensive guidance on how to turn these into preliminary system maps. For more information systems mapping methods themselves, we recommend the open access book by [Barbrook-Johnson and Penn \(2022\)](#).

Large language models

Large language models (LLMs) are probabilistic models that are capable of processing ‘natural’ language, i.e. languages used by humans. By analysing millions of text documents as part of their development (known as ‘pre-training’), LLMs have become powerful enough to competently interpret and generate human language.

LLMs generate responses to inputs known as ‘prompts’. After a human submits a prompt to the LLM, the model splits the prompt text into many smaller ‘tokens’, each consisting of a few characters (i.e. letters, spaces and punctuation). These tokens are then ‘embedded’ with semantic information gathered through pre-training. This allows the model to predict a suitable ‘response’ to the entire prompt.

An emerging application of LLMs is to prompt the model to complete a set of tasks. Such prompts might ask the LLM to perform an administrative task involving text, such as writing a cover letter, or proof-reading an email. LLMs have become rapidly more competent at these tasks over the last few years. As a result, LLMs such as *ChatGPT* have become household names.

Systems mapping for policy

Systems mapping is a suite of methods used to develop visual descriptions of how systems work. Building a system map of a topic can be useful for a range of reasons. It can help us discipline our thinking about how that system works, it can help us share and build understanding with others, and it can help start us thinking about the implications of change or intervention in that system.

There are different ways to do this, but a common form is to use networks of boxes and arrows, or ‘nodes’ and edges’ to describe the causal structure of the system. That is, to describe what factors, or variables, have some influence on others.

Systems mapping has been particularly well used in policy contexts in recent years. Maps are made of policy topics, with more or less complex descriptions of policies and their outcomes, as well as other factors affecting them. Sometimes these can look a lot like ‘Theory of Change diagrams’ or ‘logic diagrams’, and systems mapping is often explicitly undertaken to support the creation of these diagrams.

Systems mapping with LLMs

Currently, system maps are typically produced by groups of individuals coming together in meetings and workshops to construct maps together. Alternatively, individuals might read a large collection of literature on a given policy area and summarise as a list of factors and the relationships between these factors which can be used to produce a system map. In both cases, this can be an incredibly useful yet time-consuming process. This process also tends to underutilise any written knowledge on that topic, or quantitative data that may exist on that system. Here, we are interested in thinking about how we can supplement these processes by maps produced in automated but reliable ways.

LLMs may be able to help by reading and summarising policy documents and outputting a comprehensive list of systems factors and relationships in specific formats. These include but are not limited to **causes** and **effects** from causal statements. Causes and effects are individual factors, while the relationship describes the dynamics between these factors, for example:

Sample text in literature:

“The authors found that an increase in X leads to a statistically significant reduction in Y.”

Cause: X

Effect: Y

Factors (nodes) in policy system: X and Y

Relationship (edge) between X and Y: Arrow pointing from X to Y

Despite LLMs becoming increasingly powerful, systems mapping remains a demanding task which pushes LLMs to the limits of their current capabilities. In our experimentation with LLMs, we found rather quickly that image-generation models¹ (such as OpenAI's *DALL-E 3* model) are unable to produce visual systems maps to a basic usable standard. However, when restricting the request to ask the LLM to identify pairs of causes and effects from texts, it performed much better. Therefore, our research and this guidance focus on how to extract cause-and-effect pairs from LLMs and the various techniques which can improve the quality of the LLM output.

In this guidance, we aim to demystify LLMs by explaining how they work, how they can be used to better understand policy systems and highlighting their associated risks. We then provide step-by-step guidance explaining how to extract the cause-and-effect pairs and network data needed to produce a preliminary system map using a popular LLM. Importantly, we discuss 'prompting' techniques that appear to improve the output accuracy of LLMs and enclose an *example prompt* which incorporates these techniques and produced the most relevant and useful outputs during our testing.

Despite the LLM space being relatively new it continues to rapidly advance, with new language models and capabilities emerging each week. In light of this, we do not introduce this guidance as a robust method, but rather a proof of concept. We hope that introducing our findings to a wider audience will allow for true scrutiny of our method and for improvements on our method to emerge through ongoing testing with newer models.

¹ More commonly known as *diffusion* models.

Which LLM should I use?

Our recommendation: ChatGPT

*ChatGPT*² is a chat-based LLM product that is accessible through an internet browser. The product is comprised of several different large language models developed by *OpenAI*. We recommend using ChatGPT to extract cause-and-effect pairs from policy documents, since at the time of writing:

- In an independent review of the highest-quality LLMs, ChatGPT appears most often with four of its constituent LLMs appearing³.
- ChatGPT produced the most relevant and accurate responses to our early stage prompting attempts.
- Unlike some other models, ChatGPT works ‘*out of the box*’, meaning no knowledge of coding (or using an ‘*API*’) is required to use it.

Our research was conducted on the GPT-4o model, which is the default model for premium subscription accounts⁴. This model also supports file attachments, unlike OpenAI’s o1 models.

What is the difference between *ChatGPT* and *GPT-4*?

GPT-4 is a large language model and ChatGPT is the name of the chat product that is powered using the GPT-4 LLM. As a researcher, you interact with the product (ChatGPT). This product relays prompts and responses to the LLM (GPT-4) which operates in the background.

For the purposes of following this guidance, it is not necessary to make a distinction between the model and product. This is likely to apply to most proprietary LLM products, not just *ChatGPT*. Therefore, throughout this guidance we use the terms ‘*the LLM*’, *ChatGPT* and *GPT-4* interchangeably.

Is GPT-4 open source?

No, GPT-4 is a ‘proprietary’ LLM. This means its developer *OpenAI* has not made the model publicly available. Therefore, you may wish to consider alternative ‘open source’ models which offer greater transparency and control. However, these models typically require additional knowledge (including programming) to set up. We focus on *ChatGPT*

² GPT stands for *Generative Pre-trained Transformer*. Models with such generative capability are more broadly referred to as “*Generative AI*” models.

³ “ChatGPT” refers to the overall front-end product offered by *OpenAI*. This product provides access to several constituent large language models. As of December 2024, an independent ranking by *ArtificialAnalysis.ai* lists (in rank order by quality index) five ChatGPT (OpenAI) models (*o1-Preview*, *o1-Mini*, and two versions of *GPT-4o* and *GPT-4 Turbo*) among the top ten highest quality proprietary LLMs. See Appendix A for full list.

⁴ At time of writing.

for its ease of access; however, we encourage advanced users to explore the market and attempt this task with proprietary and open-source models.

Are other LLMs available?

Other high-quality LLMs include, but are not limited to: *Gemini*, *Claude*, *Mistral* and *Llama*⁵. These LLMs vary in terms of their features, capabilities, and monthly subscription fee. In addition to these general-purpose LLMs, models powering products such as OpenAI *Deep Research*, Google *NotebookLM* and Elicit may be better suited to research tasks.

The LLM landscape is rapidly evolving with performance improvements and new models with greater capabilities being released by LLM developers every week. Online leaderboards such as those published by *ArtificialAnalysis.ai* (commercially available closed-source LLMs) and *HuggingFace* (open-source LLMs) track up-to-date performance of LLMs. We recommend exploring these and any other models you may find before committing to one specific LLM.

Should I pay to use an LLM?

Like many LLMs, ChatGPT is a freemium product, where basic features are available to access for free and advanced features require a paid (premium) subscription. The free version of ChatGPT provides a limited number of requests to the newer GPT-4o model, before reverting back to an older model which produces a less accurate output. As such, it appears to be necessary to upgrade to a paid subscription for ChatGPT for the purpose of following this guidance. GPT-4o is able to handle a much larger quantity of contextual data⁶ than earlier models, meaning it is better suited to analysing long policy documents.

We restrict this recommendation to within the context of using LLMs to extract cause-and-effect pairs from policy documents for systems mapping. Furthermore, we encourage you to register for a free account and testing your own prompts to see if a paid subscription is necessary to your requirements.

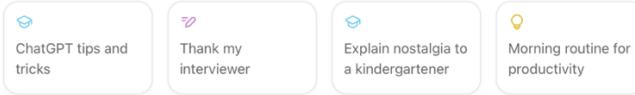
How do I create a ChatGPT account?

1. Navigate to www.chatgpt.com
2. Register for an account.
3. Once your account is successfully created and you are signed in, the ChatGPT console will show as below in Figure 1.

⁵ We identify the remaining LLM products which were included in the aforementioned independent LLM quality ranking by *ArtificialAnalysis.ai*.

⁶ You may see technical documentation refer to this as the “*context window*” of the LLM.

ChatGPT 4 ▾



Message ChatGPT



ChatGPT can make mistakes. Check important info.

Figure 1: A blank ChatGPT chat window.

Basics of how to use an LLM

While the level of technical skills required to use an LLM varies between models and products, using an LLM generally consists of the following three stages.

1) You provide the LLM with a prompt.

You begin the process by writing a “prompt”.

This is a piece of written text that is inputted into the LLM.

You write your prompt according to what you need help with. This is typically either in the form of a question or a request.

Some LLMs allow you to attach files such as PDF (.pdf) and Word (.doc) documents, comma-separated value (.csv) files, or text (.txt) files alongside your prompt as additional contextual information.

2) The LLM processes your prompt.

The LLM processes your prompt and any attached documents.

This processing happens in the “neural network” of the LLM.

3) The LLM provides you with a response.

Once the processing has finished, the LLM begins to generate a response to your prompt as its output.

This is usually in the form of a piece of text; however, some models are also capable of generating different modes of data such as images and documents.

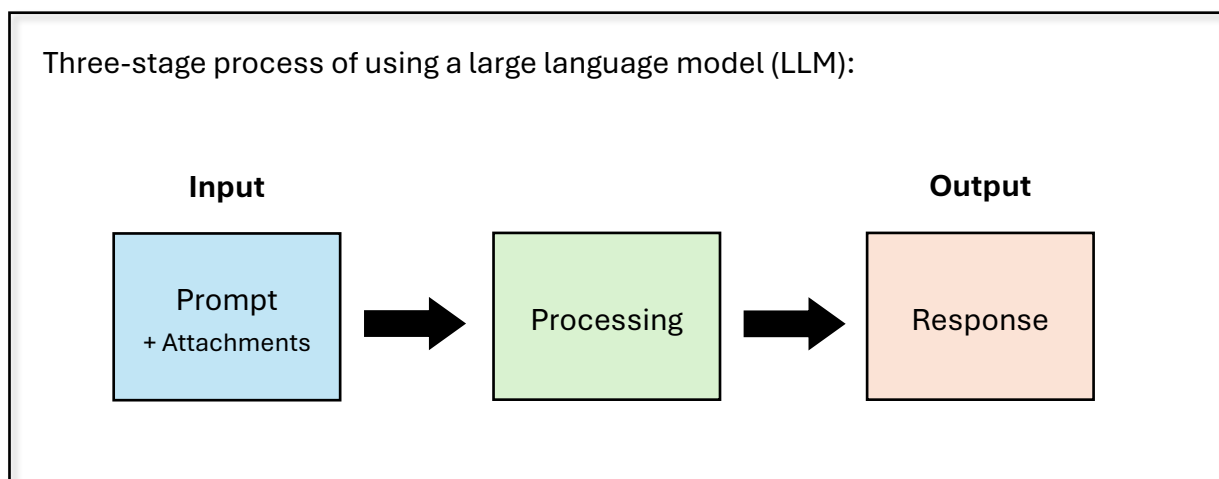


Figure 2: Three-stage process of using an LLM.

The three-stage process outlined above (Figure 2) represents how LLMs are used in general. We enhance this in subsequent sections to create a *six-stage process* of using LLMs in systems mapping.

How does an LLM work?

In simple terms, a large language model (LLM) is a piece of software which can process human or ‘natural’ language. The capabilities of an LLM emerge following a period of ‘training’, during which the model analyses millions of text documents on a wide range of subject matter. Through this process, the LLM builds up an abstract statistical pattern of the structure that human language follows. This allows the LLM to generate a response that may appear as if it was written by a human.

Simple explanation

Consider the following example⁷, where an LLM is prompted to complete a simple task.

Input (prompt):

Can you suggest one word to complete the following sentence:

We are staying indoors because it is _____.

Processing:

Likelihood of the next word being:

raining (87%) 

snowing (68%) 

sunny (12%) 

Output (response):

*We are staying indoors because it is **raining**.*

The LLM predicts that the English word “raining” is 87% likely to appear following the words “*We are staying indoors because it is...*”. Given “raining” has been assigned the highest probability score, the LLM adds this word to the sentence.

Context is key

In the above example, the LLM provided an output which is both relevant and accurate to the context. However, for more complex tasks, it will be necessary to provide the LLM

⁷ This example, including the probabilities stated, are fictitious and only intended to demonstrate how LLMs work at a simple intuitive level. In practice, variation between attempts can result in a word with a lower probability score being selected by the LLM. This can further be influenced by varying model parameters such as top-p, temperature and penalties. Additionally, this example expresses a relatively *transparent* explanation of how LLMs work. However, the processes of internal reasoning made by an LLM (through neural networks) is typically more *opaque* and appropriately referred to as a “black box” by many authors.

with more contextual information in your prompt, to ensure the output remains relevant and accurate.

If we knew that today's weather is cold but also dry, we would no longer deem "raining" as a relevant or accurate response. However, given that our prompt did not include additional contextual information about today's weather, the model did not account for this⁸. This demonstrates how an absence of context results in the output generated above instead being *accurate* and *relevant* to trends in its training data. This reliance on training data demonstrates how LLMs require a sufficient amount of context to produce relevant and accurate outputs. As we will see, it is often necessary to attempt many different prompts before a suitable output is realised.

Tasks are usually more complex than this

The above example demonstrates a simple prompt, which makes a request to the LLM to complete a simple task. Naturally, real-world use cases such as systems mapping are much more complex. As such, they require longer and more complex prompts, which require more computationally intensive analysis via the LLM.

Instead of predicting words, LLMs split a piece of text into units called "*tokens*", each of which consist of a few characters (letters, numbers, spaces and punctuation). Given the position and contextual importance of each token, the LLM successively predicts the next token (in a sequence of tokens) to eventually build up an entire response.

Despite this, we hope that our example helps to intuitively illustrate how probability-driven predictions and training data influence the response of an LLM and how this can sometimes lead to outputs that may not be contextually relevant or accurate.

⁸ As stated in [7], we discuss a simplified and fictitious scenario solely to provide an intuitive explanation of how LLMs work. Newer LLMs (including GPT-4o) are able to access some real-time data such as weather reports from the web. Hence, prompting these models with our above example may yield a more accurate response through the use of real-time weather data.

Are there any risks associated with LLMs?

Over the last couple of years, LLMs have been rapidly adopted throughout society and show potential value in most workplaces. This widespread adoption can give the impression that LLMs are generally safe to use. However, LLMs remain an emerging technology. Their performance at specific tasks, along with their associated risks are still beginning to be understood. Despite this, it is still possible to use LLMs safely as a tool for early-stage systems mapping by exercising caution and scrutinising their output.

Below, we outline a non-exhaustive list of potential risks associated with the use of LLMs for this task. In addition to our own discussion below, we recommend referring to [this](#) resource by M.J. Crockett (2025), which provides further discussion on the risks and costs of using LLMs in research.

Organisational policies

First and foremost, LLMs should be used with the same level of caution as any other digital tool. Before using an LLM in your workflow, you should consult the data protection and AI policies of your organisation. LLMs are general-purpose language processing tools and may not be designed to meet the standards and requirements of your organisational policies. You should ensure you have authorisation to use LLMs in your workflow. Your organisation may have a preferred LLM, which may be different to our recommended tool.

Decisions lie with the human expert

LLMs are technical tools which use statistical and machine learning methods to generate outputs. As tools, they can complement the work of analysts and policy makers, but they are not capable of human-like reasoning in all contexts⁹. Some authors draw caution to using anthropomorphic terms like “hallucinations” and “AI assistant” which can attribute human behaviour to LLMs. Broadly treating LLMs as equivalent and substitutable to humans may lead to an overestimation of the capability of LLMs¹⁰, an overdependence on AI tools, and ambiguity about who is accountable when things go wrong.

⁹ We make this claim to avoid the assumption that LLMs are capable of artificial general intelligence (AGI), which are hypothetical models that are capable of human-like reasoning in all tasks. Recent research in a range of fields has demonstrated that specific models are able to complete specific tasks at least as well as humans. However, this research continues to emerge and evolve as LLMs continue to be optimised and more widely adopted. For this reason, we refrain from making broader claims on human-like reasoning, in favour of policy analysts researchers applying caution when using these tools.

¹⁰ Bender et al.’s (2021) description of LLMs as *stochastic parrots* is a useful metaphor here, suggesting that much like parrots, language models repeat human language without holding a true understanding of it.

As an expert, it is the role of analysts and policy makers to exercise creative and critical judgement of the output of an LLM, which may not be able to identify cultural or contextual nuance that is otherwise recognisable to a human. It is important to remain critical of their decisions, check over their output, corroborate with other resources and use your own judgement to nullify any claims they make which you think are questionable.

For preliminary systems mapping, LLMs should be used alongside your own work, rather than used to substitute it. Any final decisions lie with you, the human expert, rather than the LLM.

Developers can train their models on your inputs

In their *Terms of Service (ToS)*, many proprietary LLM developers specify their right to retain conversational data from chats and use this to train their models in the future. In addition to your prompt text, this data may also include responses and attached documents. Developers generally require mandatory consent to their ToS upon signup, before being able to access the LLM. In theory, if any personally identifiable or commercially sensitive information is incorporated into training data, this could re-emerge in future responses given by the LLM to all users, not just you. With this in mind, it is critical to ensure that you only provide an LLM with appropriate data.

LLMs may give you false information

Since AI models operate using probability, there is always a chance that they will get things wrong. While humans are prone to making mistakes too, LLMs are programmed to always give you an answer. In such cases, LLMs may fabricate information. These fabrications (more widely known as “hallucinations”) might include seemingly factual statements and source information that appears to be genuine. In the context of systems mapping, this means there is potential for fabricated causal statements to appear among genuine cause-and-effect pairs in the LLM-generated output.

A major benefit of using LLMs in systems mapping is that analysis of the policy system can be informed by a much broader range of perspectives and understanding from secondary research data. This can help to create a more granular representation of the policy system. However, truly impactful policymaking is achieved through collaboration with the people and groups who the policy will impact¹¹. Consequently, it is important to avoid an overreliance on AI. The use of LLMs in systems mapping should supplement accurate and valuable primary research data collected from stakeholders and policy experts, rather than substitute it.

Prejudice can emerge through the output of an LLM

¹¹ CECAN have published a range of materials on *participatory systems mapping* including [this](#) toolkit.

As *generative* AI tools, LLMs are able to make *suggestions*. However, in this guidance we avoid discussion of asking the LLM to suggest additional factors for the system being analysed. Asking the model for suggestions is likely to incorporate information from outside of the context of the policy document being analysed, i.e. the model's *training data*. If this data contains material contains prejudice against specific groups, the output of the LLM could be subject to 'algorithmic bias'¹² where said prejudice is reproduced in the output. Using these algorithmically biased representations as evidence to inform policy recommendations could present potential harm to marginalised and vulnerable groups who may be misrepresented in the training data of the model.

Prompting the LLM to focus only on the target document does not omit all risk of algorithmic bias. In theory, doing this means the model needs to rely far less on its training data to generate a response. However, LLMs operate in an opaque manner, and it is often impossible to reason why a specific response was generated. Furthermore, other technical components of the LLM may be biased in addition to its training data¹³. Therefore, true omission of algorithmic bias is not guaranteed, and you should always remain cautious of this. To do so, we repeat the importance of remaining critical of the output of the LLM. The output generated by the LLM should only be used to seed further exercises which include participation from stakeholders, allowing you to build up a much more accurate representation of a policy system.

LLMs have a problem with transparency and explainability

In many cases, you may wish to reason why an LLM produced a given output. While it is possible to prompt the LLM to explain its reasoning, it is important to be mindful that any response given is generated using a probabilistic model. LLMs are trained to produce acceptable responses and this extends to questions of reasoning. In other words, it is possible that LLM will respond with an explanation that you want to hear, rather than explaining the true reasoning it used. In fact, it is often impossible to determine why an LLM has made a certain decision, with its inner workings commonly described as an opaque "black box". Therefore, it is important to bear in mind that LLMs have a poor level of transparency and explainability.

In absence of full transparency of the LLM, as a human expert, you can still adopt a policy of full transparency. This can be achieved through maintaining a research diary and retaining logs from the chat window of the LLM you are using¹⁴, along with any raw

¹² Many developers have responded to such claims by implementing policies on algorithmic bias and "guardrails" which moderate the behaviour of the LLM.

¹³ Transformer models include the use of *word embeddings*, which may also be subject to bias, as originally noted by Bolukbasi et al. (2016).

¹⁴ We explain how to do this with ChatGPT in Stage 5 of our guidance.

data that is extracted. This will help you to keep a 'paper trail' which can help to explain why decisions were made.

Guidance: How to use LLMs to extract cause-and-effect pairs from texts

This guidance explains how to use an LLM to extract cause-and-effect pairs from texts for preliminary systems mapping. The guidance is developed around ChatGPT, but we expect the process to be similar for other web-based LLM products.

This guidance is designed to be a basic introduction to using LLMs. As such, we do not discuss using LLMs in more complex ways (such as via an API¹⁵).

Using an LLM as part of the systems mapping process

Earlier in this guidance, we outlined the three core stages of using an LLM. We build on this with three additional core stages necessary to the specific context of systems mapping. Hence, this guidance is structured around the following six core stages:

1) **Prepare the policy document.**

To improve the accuracy of the output, you convert your policy document to plain-text format.

2) **Provide the LLM with a prompt and the policy document.**

You write a prompt requesting the LLM to complete cause-pair extraction task.

You submit this prompt to the LLM along with the plain-text policy document.

You test and refine your prompt as necessary.

3) **The LLM processes your prompt.**

The LLM processes your prompt and document and extracts cause-and-effect pairs.

4) **The LLM provides a response.**

The LLM generates a list of cause-and-effect pairs as its output.

5) **Export the list of cause-and-effect pairs.**

You export the generated list of cause-and-effect pairs for use in systems mapping software.

¹⁵ Application Programming Interface (API) setup often requires knowledge of programming language (e.g. *Python*) and API access may vary between models and developers. However, [this Medium article](#) provides simple instructions on setting up the OpenAI API. Please note that API access is associated with additional risks that are not discussed in this guidance.

6) **Create a visualisation of the policy system.**

Using the exported list, you create a visualisation (systems map) of the policy system.

Figure 3: Six-stage procedure of using large language models (LLMs) to extract causal statements from texts for systems mapping.

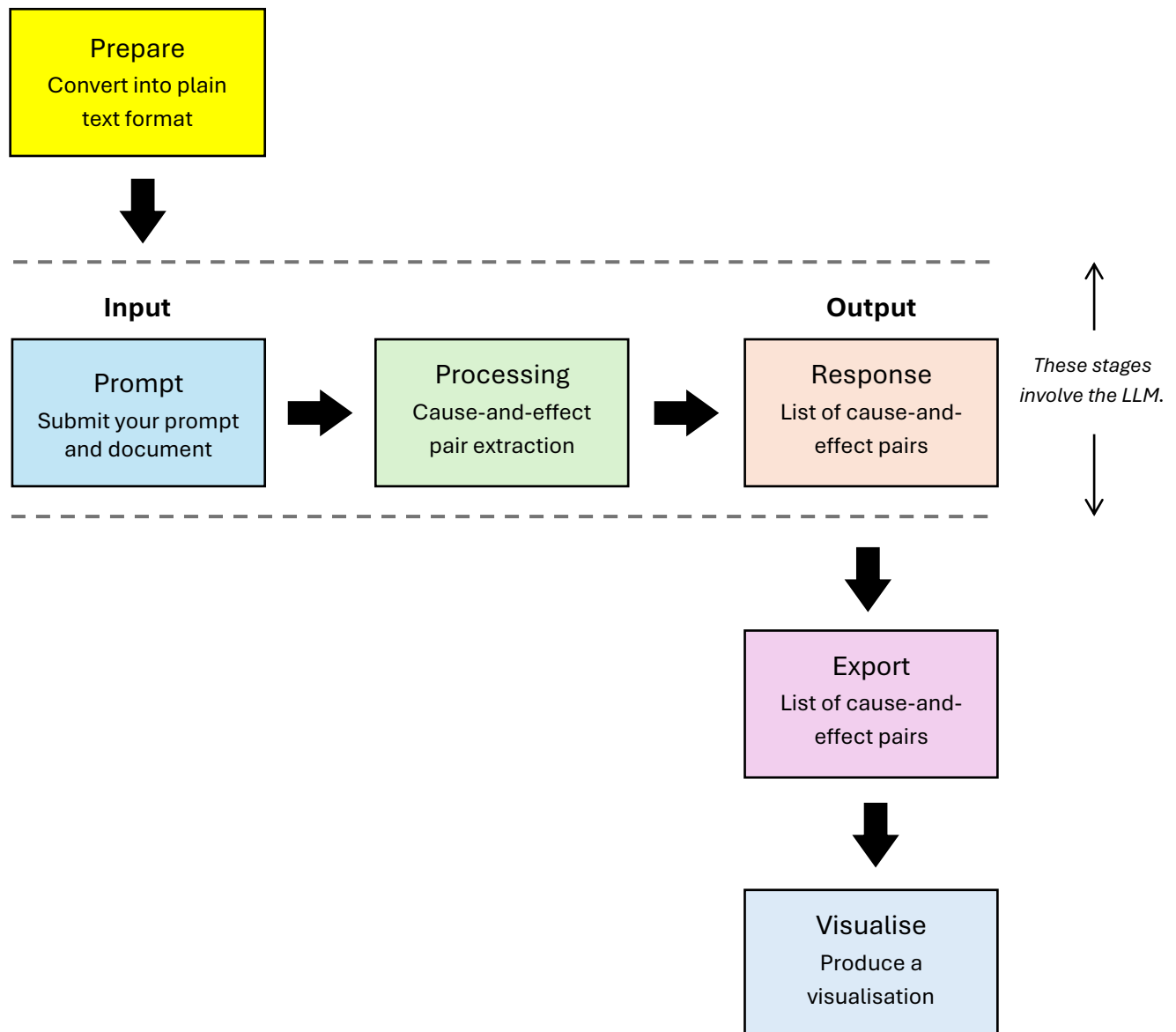


Figure 3: Our six-stage procedure for using LLMs in systems mapping.

1) Prepare the policy document

Converting a document into plain-text format

LLMs generally have a limit to the amount of information they can process at any one time. Documents such as policy briefs, articles and academic papers can be visually complex and include different fonts, tables, graphs and images, which may constrain the LLM's accuracy at extracting cause-and-effect pairs from the text.

These are most commonly in the form of PDF (.pdf), Word (.doc/.docx) or HTML (.htm/.html) documents. During our testing, we found that converting files to “**plain text**” substantially improved the quality of the output from the LLM. In terms of the quantity of cause-and-effect pairs identified, on numerous occasions this doubled when compared to the output when uploading a document in its original PDF format.

Converting to plain text means removing any formatting from a document or piece of text.

*This text contains formatting such as **bold text**, colour, and different fonts.*

This text contains no formatting.

Below, we explain how to use Microsoft Word to convert a PDF document into plain text. Other word processing applications are likely to have similar capabilities, but native applications such as *Notepad* for Windows, or *TextEdit* for Mac OS do not appear to work well for this task and should be avoided. The below process must be repeated for each individual policy document.

Importantly, this process only works for documents containing highlightable text. Some older documents, or those which have been scanned from paper copies, may not contain highlightable text. Some institutions will have access to Adobe Acrobat which is also suitable and includes capability of converting such documents into plain text (known as OCR).

Converting a PDF document into plain text using Microsoft Word:

1. Open Microsoft Word.
2. Go to File > Open
3. Navigate to the document you wish to convert and click Open.
4. Word should begin to automatically convert the file into *Word* format. This is not yet in plain text format.
5. Go to File > Save a Copy... (formerly “Save As ...”)
6. Navigate to the folder you wish to save the plain text version of the document in.
7. Under *File Format*, select **plain text (.txt)**
8. Save the file in a convenient location.

9. A '*File Conversion*' dialog box may appear with some options for customisation. A plain text version of the document will be previewed at the bottom of this box. Confirm that the text is legible and click OK to continue.
10. If a subsequent dialog box appears, click Yes to continue.
11. The document will be saved in plain-text format.
12. Repeat these steps for each individual policy document.

Word may sometimes struggle to convert files. An error message may be displayed, or text within the document may show incorrectly such as a loss of spaces between words. In the first instance, try repeating the above steps again. If this does not work:

- a. Open the original document.
- b. Go to Edit > Select All to highlight all text within the document.
- c. Go to Edit > Copy to copy the highlighted text.
- d. Navigate to Microsoft Word and create a new blank document.
- e. Go to Edit > Paste to paste the text.
- f. Follow Step 5 onwards from above to save as plain text.

Can the conversion to plain text be automated using an LLM?

Given that LLMs are useful in administrative tasks, it is reasonable to assume that they may be able to convert documents into plain text too. However, we found that GPT-4o was unable to convert PDFs into plain text, as the file produced was blank each time. Therefore, manual conversion of policy documents is still recommended at this time.

What about loss of images and diagrams?

A policy document in PDF format may be 'rich' with text, images, and diagrams. Converting into plain text format will only retain text from the document. The structure of any diagrams containing highlightable text will be lost, along with any non-highlightable text and images. Losing this information is offset with the substantial increase in output quality observed by converting into plain text format.

Work with one document at a time

LLMs can only process a limited amount of contextual data. During our testing, we found that ChatGPT was able to generate a much higher quality output when analysing one paper at a time rather than submitting multiple documents alongside the prompt. Therefore, we recommend only uploading one paper alongside your prompt and repeating the full process for each additional paper. This helps to avoid submitting too much contextual data to the LLM.

2) Provide the LLM with a prompt and a policy document.

To extract cause-and-effect pairs from the source document correctly, the LLM needs to be instructed to do this using a suitable prompt. This prompt needs to encompass the specifics of this task in a comprehensive yet succinct manner which requires a delicate balance to achieve. Furthermore, it is likely that no single ‘optimal’ prompt exists. Instead, prompts can be written in countless different ways while still achieving the same objective. Hence, this process requires writing a prompt from scratch and iteratively developing it with successive refinements until you find a prompt that works to an acceptable level. This process is widely referred to as “prompt engineering”.

To support your own prompt engineering, we outline below the techniques that we found improved the accuracy and relevancy of the output from *ChatGPT* during our testing. This is followed by our *example prompt*, which incorporates all of these techniques.

Testing and refining your prompt

LLMs are built upon probability, meaning *ChatGPT* is unlikely to generate identical responses each time you run an identical prompt. The list of cause-and-effect pairs generated may vary each time you submit the same prompt and document to ChatGPT. In other words, there is always some level of variability in the output. Therefore, we recommend testing your prompt by submitting it multiple times, each time in a new chat window. This will allow you gain your own intuition of how much the quantity and quality of the output can vary between attempts. This intuition can be extremely informative when making refinements to your prompt. As the capability of LLMs continues to develop rapidly, you will likely need to refine the wording of your prompts on an ongoing basis to ensure they are adapted to newly released models.

Prompting techniques

Prompt Sandwich 🥪

A common metaphor among prompt engineering literature is the concept of a *prompt sandwich*. Essentially, your prompt should be structured with three distinct layers*, much like the bread and filling of a sandwich. In order, these layers are:

- **Steer the conversation and role of the LLM**
- **Provide context**
- **Make a request**

(Adapted from Macro and Spong, 2023)

Your prompt should begin by explaining the purpose of your interaction with the LLM and assigning roles. Next, you should provide additional context to narrow the focus. Finally, you should make a request by introducing and outlining the specific tasks that you require the LLM to complete.

**For systems mapping, most of the text in your prompt is likely to be dedicated to the final step of making a request. Nevertheless, each layer is distinctly important, and we still recommend using the above layers as a structure when designing your own prompt.*

Simple prompt

A simple prompt might look like this:

Can you read the attached file and give me a list of causes and effects.

From one perspective, this prompt is extremely concise. It contains nothing unnecessary and is straight to the point. The LLM ‘knows’ what task is required and can process this quickly. However, this prompt fails to provide the LLM with any specific requirements or preferences regarding how we would like the task to be completed. As a result, ChatGPT responds to this prompt by providing an output list in a generic format. When we submitted this prompt, ChatGPT identified four causes and eight effects, but provided no information about how these causes and effects were connected.

Identify yourself

The output of the LLM can be strengthened by identifying an audience for the output (Technical Life, 2024). This helps to prime the LLM (Spyscape, 2024) to focus on producing an output which is aligned with your area of expertise.

I am an expert in _____ policy. I need you to...

Naturally, the audience for the cause-and-effect pair list is going to be you. Identifying yourself as an expert of a specific policy area relevant to the document you are submitting can steer the LLM toward generating language which is more specialised and appropriate to that policy topic. When we omitted this line from our prompt, we observed a slight decrease in the quantity of pairs identified by the LLM.

Conciseness

A simple prompt can give the LLM too much freedom, meaning the task is not completed in the way we need it to be. Adding more information to the prompt is helpful, but this needs to be done concisely.

Use consistent vocabulary

When referring to the same concept more than once in your prompt, ensure that the same vocabulary is used throughout.

Can you read the attached document and give me a list of causes and effects. You must only give me pairs which are described in the file.

This prompt refers to cause-and-effect pairs as “causes and effects” and “pairs”, and the text as “document” and “file”.

This imprecision can lead the LLM to make mistakes as it may not understand that we are referring to the same concepts in either of these cases.

Ensuring that the same vocabulary is used reduces the likelihood of misinterpretations by the LLM.

Can you read the attached document and give me a list of causes and-effects. You must only give me **causes and effects** which are described in the **document**.

Be positive but impolite

To make your prompt more concise, avoid being polite and using superfluous words such as “please” and “thank you”. Omitting this type of language focuses the prompt on the core task in-hand (Technical Life, 2024).

Be precise

When referring to specific concepts, use precise language.

Can you read the attached document and give me a list of causes and effects. You must only give me causes and effects which are described in the document.

In the second sentence above, “the document” is used. This is ambiguous as you could be referring to a different document. Be clear that you are referring to the same item.

Can you read the attached document and give me a list of causes and effects. You must only give me causes and effects which are described in **the attached document**.

Use concise language

There is a delicate balance between a shorter prompt which is quicker to process but more ambiguous, and a longer prompt which is more precise but takes longer to process. Balancing this effectively is down to your judgement.

The attached document relates to decarbonisation policy and discusses electric vehicle (EV) deployment in the United Kingdom.

The above text does a good job of summarising what the attached document is about. If we need to refer to the document again, referring to it as “the attached document” is likely to suffice.

The attached document relates to decarbonisation policy and discusses electric vehicle (EV) deployment in the United Kingdom.

...

Can you read **the attached document** and give me a list of causes and effects.

As your prompt becomes longer and more complex, it is important to look through it and remove any superfluous text. Try to reduce the number of words in your prompt as far as possible without losing meaning or increasing ambiguity.

When not to be concise

In some cases, it may be favourable to repeat specific statements to indicate their importance.

We found this was useful where you have one specific rule that you need the LLM to adhere to. In our initial testing, introducing a rule and repeating it again at the end of the prompt resulted in this rule being adhered to more frequently.

If you have multiple rules which you need the LLM to abide by, we recommend a different approach which is discussed later under “Rules”.

Providing instructions

Your prompt should provide the LLM with clear instructions on what task is required and what steps are necessary to complete it.

Task delineation

Breaking a complex task down into individual steps can make it easier for the LLM to follow (Technical Life, 2024). Instead of one block of text which explains the task, include a separate section of your prompt which delineates exactly how the task should

be completed, step by step. Use a numbered list to show the order in which each step should be completed.

A task is a summary of the overall process and output that we want the LLM to achieve but may not be entirely precise. For systems mapping, a task might be:

Task (i): *Analyse the attached document and create a list of cause and effects based on all causal statements.*

To complete this task, we need the LLM to follow a set of steps. These are the specific consecutive stages of the process that the LLM needs to follow to achieve this task. For the above task, we can delineate these steps in the prompt as shown below.

I am an expert...

...

Can you complete the following steps:

- 1) Read the attached document.
- 2) Provide me with a list of causes and effects.

Ensure the final step explicitly asks the LLM to generate the output

Ask the LLM to generate the output as the final step. This ensures that generation does not occur before it has completed any critical steps in completion of the task. Without this, the LLM may begin generating text before it has analysed any text. In our testing, this typically led to far fewer pairs being identified and a poor-quality output.

Rules

Include a list of rules that the LLM must stick to, to achieve the task successfully. In a similar way to task delineation, use a bulleted list to identify specific rules which the LLM must follow when completing the task. Explicitly state that these rules must be abided by.

Set a goal

Include a specific number of pairs which you think is reasonable to aim for, for your document. When testing, we expected around 50 cause-and-effect pairs to be extracted from our first document. By specifically stating this, we created a benchmark for the LLM to aim for. Instead of stopping after identifying around 25 pairs, the LLM continued to identify pairs until it reached our expectation. Frequently, it exceeded this level.

It is important to set a useful and realistic goal. Asking the LLM to identify 300 cause-and-effect pairs for a 10-page research paper is unrealistic and is likely to lead to spurious output being generated.

As discussed above, we included an expectation of 50 cause-and-effect pairs for one of our sample documents. When increasing this expectation from 50 to 75 pairs, we saw the quantity of pairs decrease by roughly 30%. Therefore, it is worth varying this until you find an acceptable expectation.

Tell the LLM to avoid rounding

ChatGPT appears to favour rounding to the nearest five or ten. Before we addressed this, the output would often include 10 causes and 10 effects when our documents evidently contained a far higher quantity of cause-and-effect statements. ChatGPT may round to 5, 10, 20, 30 and so on.

To avoid this, state explicitly that the LLM must not round to a neat number. This is complemented by the request to work exhaustively below.

Tell the LLM to work exhaustively

The LLM may not understand the depth of analysis that is required for this task. Explain to the LLM that the list of cause-and-effect pairs needs to be exhaustive, implying that it needs to reference any and all causal statements in the text.

This complements the request to avoid rounding above.

Ask the LLM to wait until the final step to generate the output

Include a rule which explicitly states that the LLM must not generate an output until it has completed the penultimate (second-to-last) step.

...

You must take the following steps to complete this task.

- 1) ...
- 2) ...
- 3) Generate the output list of cause-and-effect pairs.

...

You must abide by the following rules when completing this task.

- a) ...
- b) ...
- c) **You must not generate an output until after Step 2 has been completed.**

In this example, the task is delineated as three individual steps which the LLM needs to follow. The final step (Step 3) asks for the output to be generated. The list of rules requires that the output is not generated until after the penultimate task (Step 2).

Tabular form

ChatGPT appears to be programmed generate text in a format which is visually pleasing – including headers, indentation and bold text.

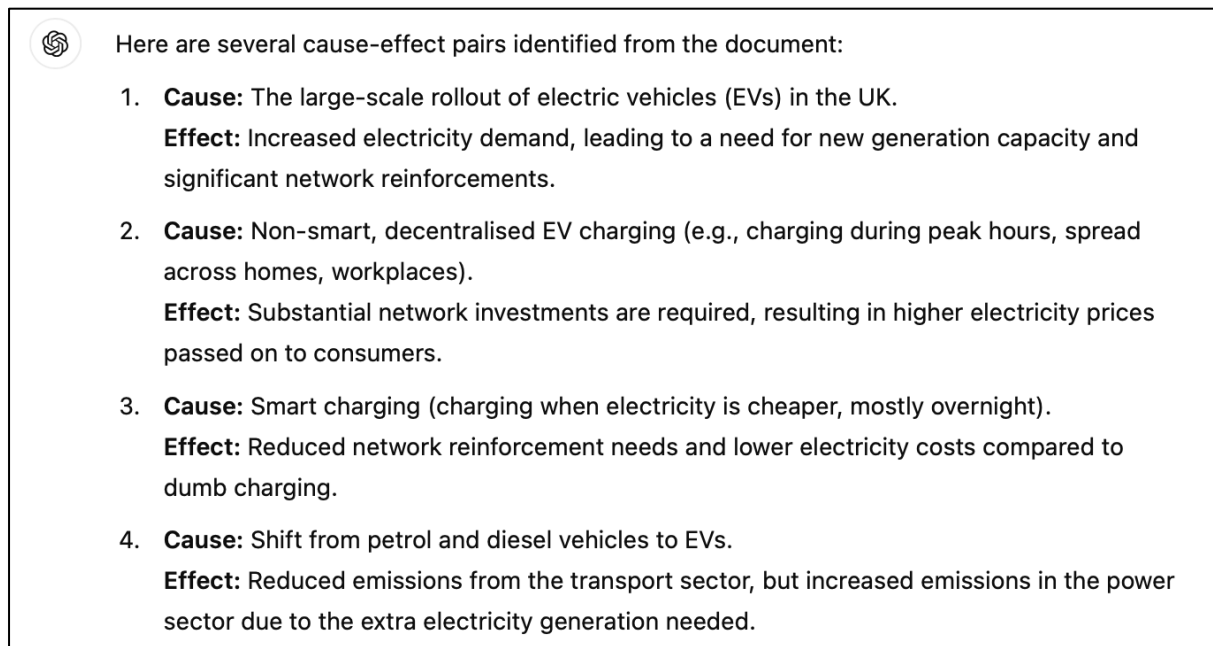


Figure 4: ChatGPT generating an output using its standard formatting.

However, the quality of the output is almost always much poorer in this format. This can lead to rounding to a neat number, a small number of cause-and-effect pairs being identified, and sometimes more causes than effects, or vice-versa.

To avoid this, include a request for the output list of cause-and-effect pairs to be generated in *tabular form*.

I am an expert...

...

The response output should be generated as a list of cause-and-effect pairs in tabular form.

This means the list will be generated as a table which contains rows, columns, and column headers.



I have thoroughly reviewed the document you provided. Here is a table that lists the cause-and-effect pairs I identified, along with the corresponding source text:

Cause	Effect	Source Text
Electrification of transport	Increased load on the electricity network	"the new electric load to charge EVs is likely to require new generation capacity" (1)
Non-smart/decentralized charging	Larger network investments required	"non-smart and/or decentralized charging will require considerably larger investments" (1)
Network reinforcement costs	Increased electricity prices	"network reinforcement costs are passed to the consumer via increased electricity prices" (1)
Large-scale EV penetration	Shift of emissions from transport to the power sector	"emissions shift from the transport to the power sector" (1)
EV charging strategies	Mitigation or increase in	"how different EV charging strategies increase

Figure 5: Tabular-form list generated by *GPT-4o*.

Asking ChatGPT to provide a source

To better identify whether generated cause-and-effect pairs are accurate, we recommend adding a request for source text for each pair. ChatGPT will provide a copy of the sentence or paragraph containing the causal statement. This approach was inspired by Powell and Caldas Cabral (2023).

You must provide the source text for each cause-and-effect pair.

There does appear to be a trade-off with this technique. In our testing, asking the LLM to provide sources for each pair reduced the overall quantity of pairs identified, sometimes by a half. However, every pair was sufficiently validated with an accurate source quote. We retained this technique as part of our prompt, as is likely to be valuable as a way to quickly verify that a generated cause-and-effect pair is genuine.

Interestingly, as part of this process we noted that GPT-4o abridged the source text by reducing a paragraph with multiple sentences, into one single sentence which maintained the same message.

Original text:

Extract from Calvillo and Turner (2020) discussing Pudjianto et al. (2013):

"Pudjianto et al. [14] develop a range of numerical simulations based on different distribution network topologies (urban and rural) in the UK, assessing the need and the cost of network reinforcements required to accommodate the electrification of transport and heat. They conclude that under current passive distribution network and demand, the electricity peak is likely to increase up to 2–3 times, and that significant distribution network reinforcement will be required."

Source text from LLM output:

AI-amended extract from Calvillo and Turner (2020) discussing Pudjianto et al. (2013).

"Pudjianto et al. (14) conclude that under current passive distribution network and demand, the electricity peak is likely to increase up to 2-3 times, and that significant distribution network reinforcement will be required."

Using multiple prompts

Prompts that include a list of tasks along with specific rules to follow can quickly become long and lose brevity. To remain concise, it may be beneficial to create multiple prompts based on each of the core tasks that are required, which are then submitted in succession. Note that we are referring to individual tasks here, rather than the constituent steps that are required for each task to be completed.

Given that the output can vary between each attempt, it may be worth prompting the LLM to analyse the paper a second time to see if any cause-and-effect pairs were missed. You can use a subsequent prompt which is submitted after the output has been generated to do this. Crucially, this second prompt followed the same structure and incorporated the same techniques as our first prompt, to ensure the output was generated to the same standard.

During our testing, we found that prompting ChatGPT to analyse the document and extract any additional cause-and-effect pairs after it had generated an output led to a higher number of cause-and-effect pairs being identified.

Avoid creating a new list and instead, append new pairs to the original tabular list

Initially, we prompted ChatGPT to read our attached paper again and create a new tabular list containing any additional cause-and-effect pairs it had found. However, the new list contained many cause-and-effect pairs that were already present in the original list. Therefore, we recommend avoiding requesting a new list for new pairs.

Instead, we prompted the LLM to add any new pairs to the list it had previously generated. In this case, ChatGPT successfully appended more pairs to the original list without losing or repeating any pairs. In this scenario, we explicitly referenced the *previously generated list* within our prompt. We suspect this led to an improvement as the LLM may have then generated an output contextually based on the original list, which helped to avoid any repetition.

Example prompt

In this section, we introduce an example text which can be used to prompt an LLM to analyse and extract causal statements as pairs for the purpose of systems mapping. This prompt led to outputs that we felt were most reliable and useful to seed the creation of a preliminary system map. The prompt contains all of the strategies which were successful in *encouraging* the LLM to produce a high quantity of cause-and-effect pairs which were relevant to the literature used.

Note of caution

We have noticed that factors relevant to our main test document¹⁶ sometimes appear in the generated output for other documents, specifically when using the below example prompt. This may suggest that GPT-4 has associated our example prompt and the content of our main test document. For this reason, we advise against using our example prompt below and instead recommend composing your own prompt using the techniques we have outlined above.

Following the *prompt sandwich* metaphor, we begin by establishing identity, then provide context of the task required, before making a request which includes the steps required to complete the task and rules to follow.

Two prompts are provided. The first prompt instructs the LLM to read the document, identify causal statements, and extract them by creating a tabular list. The second prompt instructs the LLM to repeat this method and create a *new* tabular list which captures any additional causal statements which were not previously identified. Both prompts incorporate lists of individual steps that the LLM needs to take, along with rules which must be followed.

Prompt 1: Analysis and extraction of cause-and-effect pairs

¹⁶ Calvillo and Turner (2020), as discussed under [Methods and Limitations](#).

I am an expert in _____ policy.

I need you to read the attached paper and list any cause-and-effect pairs, which are two factors relating to decarbonisation policy connected by a causal relationship.

To complete this task, you must follow these steps:

- 1) Read the paper in full and identify all cause-and-effect pairs.
- 2) Provide your response output as an aggregated list of cause-and-effect pairs in tabular form. You must include a column which shows the source text in which you found a given cause-and-effect pair.

You must abide by these rules:

- a) You must not generate a response until after Step 1.
- b) Do not round the list to a neat number of cause-and-effect pairs. You must be thorough, and the list must be extensive.
- c) You must only include cause-and-effect pairs contained in the attached paper.
- d) You must provide the source text for each cause-and-effect pair.

I am expecting you to identify around 50 cause-and-effect pairs in total.

Prompt 2: Identification of additional cause-and-effect pairs

To improve the quality of the list, I now require you to complete the following steps:

- 1) Read the paper in full.
- 2) Generate a new list containing any cause-and-effect pairs which were not included in your original list. The new list must be in tabular form.

You must abide by these rules:

- a) You must not generate a response until after Step 1.
- b) Do not round the list to a neat number of cause-and-effect pairs. You must be thorough, and the list must be extensive.
- c) You must only include cause-and-effect pairs contained within the attached paper.
- d) You must provide the source text for any new cause-and-effect pairs.

Designing this prompt

The above prompt emerged from our successive testing of each of the previously listed techniques. Starting with a simple prompt, we changed the prompt text to incorporate a given technique. The updated prompt was submitted to ChatGPT, and the generated output was inspected, to see whether an improvement was made.

✅ If the output (as a list of cause-and-effect pairs) improved, the technique was deemed suitable for this task, and we accepted the new prompt. This new prompt would be passed through to the next test. Each of the previously listed techniques were deemed suitable as their tests were successful.

❌ If the output worsened, the technique was deemed unsuitable, and we rejected the new prompt. The original prompt would be passed through to the next test.

This approach allowed us to *iteratively* design the below prompt, meaning it became gradually more complex as we tested each technique, with the below prompt emerging at the end of our testing.

We elaborate on this method under the *Method and Limitations* section.

Unsuccessful techniques

When including the following techniques in our prompt, we observed a decline in the quality of the output generated by ChatGPT.

- **Prompting for a “systems map” to be generated**
 - ChatGPT has the ability to generate systems maps in the form of *directed graphs*¹⁷. However, a simple prompt for ChatGPT to generate a system map can result in few factors being identified due to lack of brevity of the prompt as discussed earlier under ‘Simple prompt’. We also tried adding a third-stage prompt to our *example prompt*, asking ChatGPT to generate a system map. However, most of the system factors were not connected (Figure B5), meaning further manipulation of the map was still necessary, which would likely require programming knowledge to achieve¹⁸.
- **Assigning a role to the LLM**

¹⁷ ChatGPT leverages the Python programming language to generate non-textual elements including tables and charts from the textual output of its language model. To generate a system map visualisation, ChatGPT calls upon the *Matplotlib* and *NetworkX* libraries in Python which are used to create data visualisations and network graphs.

¹⁸ While it may be possible to use subsequent prompts to change simple parameters of these maps (e.g. colours and font faces), comprehensive manipulation is likely to require some working knowledge of Python vocabulary and syntax. As this guidance is targeted at a wider non-technical audience, we instead detail how to use Excel and PRSM as more widely accessible tools that can be used to achieve the same result.

- Recent literature (Mizrahi and Serfaty, 2024) suggests assigning a role to the LLM as well as yourself. For example, “*You* are an expert in agricultural policy.” When amending our prompt to include this, we found no noticeable change in the quality or quantity of cause-and-effect pairs identified. The remainder of our tests were conducted using “I am an expert” in our prompts.
- **Asking ChatGPT to count the number of cause-and-effect pairs it had identified**
 - When we asked ChatGPT to do this as part of a subsequent prompt, it almost always failed to give the correct total quantity of pairs from the tabular list it had generated, despite this being a seemingly straightforward task. Therefore, if you require this value, we recommend calculating the quantity of cause-and-effect pairs manually.
- **Providing examples**
 - Prompt engineering literature frequently introduces the idea of providing LLMs with examples to improve the quality of the output. This is commonly referred to as “*one-shot*” or “*few-shot*” prompting. However, we found that the quantity of cause-and-effect pairs identified became more inconsistent between attempts when we included an example.
 - Furthermore, ChatGPT already appears to do a reasonably good job of identifying and summarising causal statements from text. As such, providing an example of a cause-and-effect pair may make your prompt longer at the expense of additional context which may not be particularly useful.
- **Asking the LLM to justify its output**
 - Throughout our testing, we found that querying the output with ChatGPT often led the LLM to regenerate its output, where this wasn’t necessarily required, without a clear explanation. Instead, it appeared that ChatGPT may be programmed to immediately agree when its output is placed under scrutiny, rather than providing the justification of its output that we were actually seeking.
- **Removing references from the original text**
 - Giabbanelli and Witkowicz (2024) suggest removing reference lists from the plain text version of the source document. We found that this made no noticeable difference in the number of pairs identified and the generated output for both versions (baseline and with references removed) contained very similar pairs.
- **Uploading PDF documents**
 - As discussed earlier, converting PDF documents into plain text substantially improved the output.

- **Uploading more than one document at a time**
 - As discussed earlier, uploading one document at a time substantially improved the output when compared to uploading multiple documents at once.

Submitting your prompt and document to the LLM

Once your document has been prepared and you have written your prompt:

1. Open a new chat window in ChatGPT.
2. Enter your prompt into the input field. This usually says “Message ChatGPT” when it is blank.
3. Attach the text you would like to be analysed by clicking the paperclip icon and navigating to the plain text (.txt) file created earlier.
4. Click the upward arrow to send this request to ChatGPT.
 - a. If the upward arrow is greyed out, it is likely your file is still uploading. Wait 10 seconds and then try again. If this persists, refresh the page and start again from Step 1.
5. ChatGPT will process the request and respond by saying it is “Extracting text” and “Analysing...”.
6. ChatGPT generates the response.

If your prompt is split into sections which need to be processed separately, run the first prompt, and then wait until the response has been generated before submitting the second prompt in the same chat window.

3) The LLM processes your prompt.

After submitting your prompt successfully, ChatGPT will indicate that the text from the document is being extracted and analysed.

Troubleshooting

The output generation is extremely slow or stops altogether

The amount of time for ChatGPT to generate a response can vary. During our testing, this generally took no more than 30 seconds; however, you may find different results. If the output is generating extremely slowly or stops, refresh the page and submit the prompt and document again in a new chat window.

4) The LLM provides a response.

Once the LLM has finished analysing the policy document, it will generate a response.

Troubleshooting

ChatGPT can sometimes fail to produce an output. In all of the cases listed below, you should **start a new chat and submit your prompt and document again**.

Spurious Responses

At the time of publishing, GPT-4o sometimes treats the expected number of pairs in our example prompt as a quota. In such cases, the LLM generates an initial list using the text in the paper and will subsequently generate another list to meet the quota. This second list appeared to generally contain acceptable cause-and-effect pairs.

Timeout when analysing

ChatGPT will begin to analyse the document you have attached and display “Analysing...” in the chat window. Infrequently, the process will freeze at this stage with nothing else happening.

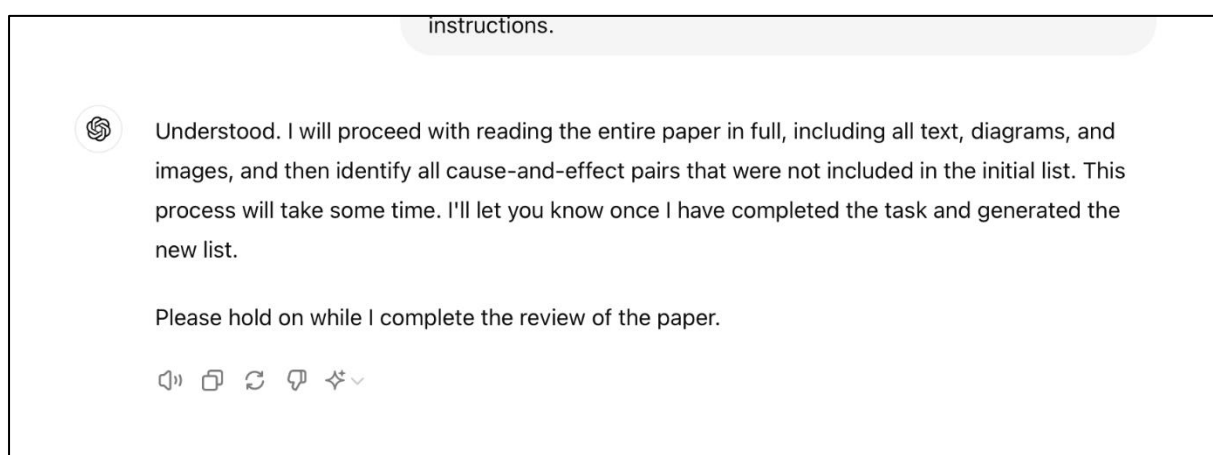


Figure 6: ChatGPT fails to analyse the uploaded document.

Failure to produce a list

ChatGPT sometimes fails to start this task. We received the following response when attempting to run our prompt.

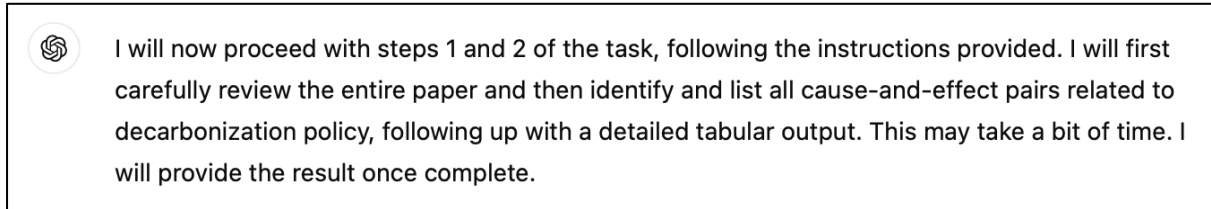


Figure 7: ChatGPT fails to generate a list.

Here, ChatGPT has avoided analysing the paper and proceeded to immediately generate a response. As a result, no list is generated. Only the above paragraph is displayed.

List is not in tabular form

Infrequently, ChatGPT will ignore your request for the list to be in tabular form. The list may be displayed with headers and bullet points, as previously shown in Figure 5.

5) Export the list of cause-and-effect pairs.

If successful, ChatGPT should have generated a list in tabular form. Tabular form makes the list extremely easy to copy into other applications such as Microsoft Excel, for conversion into other formats for use with specialist systems mapping software.

Retain a copy of the entire conversation

As discussed earlier, one associated risk of using an LLM is a lack of transparency and explainability in their decision making. To mitigate against this risk, we recommend maintaining a record of the full conversation you have had with ChatGPT.

To retain a copy of the conversation:

1. Use **Ctrl+A** (**Cmd+A** on Mac) on your keyboard to highlight all of the text in the browser window.
2. Use **Ctrl+C** (**Cmd+C** on Mac) to copy the highlighted text.
3. Create a new document in Microsoft Word.
4. Paste the text into the new document.
5. Save the document. We recommend including the name of the policy document and “ChatGPT conversation” as part of the file name.

To export the list from ChatGPT:

1. Ensure the list has been provided in tabular form as below.

Cause	Effect	Source Text
Cause name will be shown here	Effect name will be shown here	"The source text will be shown in quotation marks like this."

Figure 8: Output list appearing in tabular form.

2. Click and drag to highlight the table, including the column names.
3. Go to **Edit > Copy** to copy the highlighted table¹⁹.
4. In a spreadsheet editor such as Microsoft Excel, create a new workbook.
5. In the blank workbook, ensure cell A1 is selected.
6. Go to **Edit > Paste** to paste the table.
7. Save the spreadsheet. We recommend including the name of the policy document and “raw data” as part of the file name.

Can this be automated using an LLM?

¹⁹ A download button sometimes appears to the right of the title of the table, allowing the table to be downloaded in .csv format. Note this .csv file will need to be subsequently saved into .xlsx format to retain any Excel formatting.

As with converting documents into plain text, we recommend exporting the list manually. Some LLMs claim to be able to convert the list into different formats for you automatically. However, we found that this process was highly unreliable with ChatGPT often losing cause-and-effect pairs or failing to convert the list.

Cleaning the exported list

Data cleaning is likely to be necessary on the list generated by the LLM. Firstly, this is to ensure that this data has been verified by a human expert. Secondly, ChatGPT appears to struggle with using the same naming convention for a given factor (cause, or effect) throughout the list. This can cause particular issues with systems mapping software, as multiple nodes (factors) will appear within the map for one single factor of the policy system. Furthermore, it is crucial to perform data cleaning at this stage, as it can become much more complicated as you begin to structure the data for use in systems mapping software.

Example:

System factor: UK petroleum fuel consumption

Relevant cause-and-effect pairs generated by ChatGPT:

Cause: Change in petrol consumption in the UK

Effect: Change in fuel prices

Cause: Higher aggregate consumption of petroleum fuel

Effect: Increase in greenhouse gas emissions

Cause: Decrease in pump prices

Effect: Increase in UK petrol consumption

Unique labels for this system factor:

(1) Increased petrol consumption in the UK

(2) Higher aggregate consumption of petroleum fuel

(3) Increase in UK petrol consumption

In this example, the LLM has identified three causal relationships relating to an increase in petrol consumption in the UK. Note that in the first two pairs, the system factor is the *cause*, while in the final pair it is the *effect*. It should be reasonable to argue that each of these pairs is referring to the same conceptual idea, hence, the same system factor. In which case, we would expect the LLM to use the exact same label for this factor across all three pairs. However, it appears that ChatGPT does not regularly homogenise factor labels in this way.

You should ensure the exported list is cleaned and that cause-and-effect pairs appear relevant (using the “source text column” and employ the same naming convention. You may wish to use Excel functionality such as *Data Validation* to achieve this in a more rigid manner – this is beyond the scope of our guidance. Note that you may delete the source text information if it is no longer required, as this is predominantly for proof-reading the generated list.

6) Creating a system map using PRSM

Systems maps can be produced in various ways and there are many different software packages and tools available to help you do this.

We recommend using *Participatory Systems Mapper (PRSM)*, as it is simple to use, has been developed with systems mapping in mind, and requires no knowledge of computer programming. Additionally, PRSM can be accessed via an internet browser, without the need for software installation.

PRSM was developed by Nigel Gilbert from CECAN at the University of Surrey. PRSM is available at: www.prsm.uk

Example system map

The following system map (Figure 9) is a PRSM visualisation of the electric vehicle (EV) adoption in the UK using cause-and-effect pairs generated using ChatGPT²⁰. This map was generated using the steps and example prompt we have outlined in this guidance.

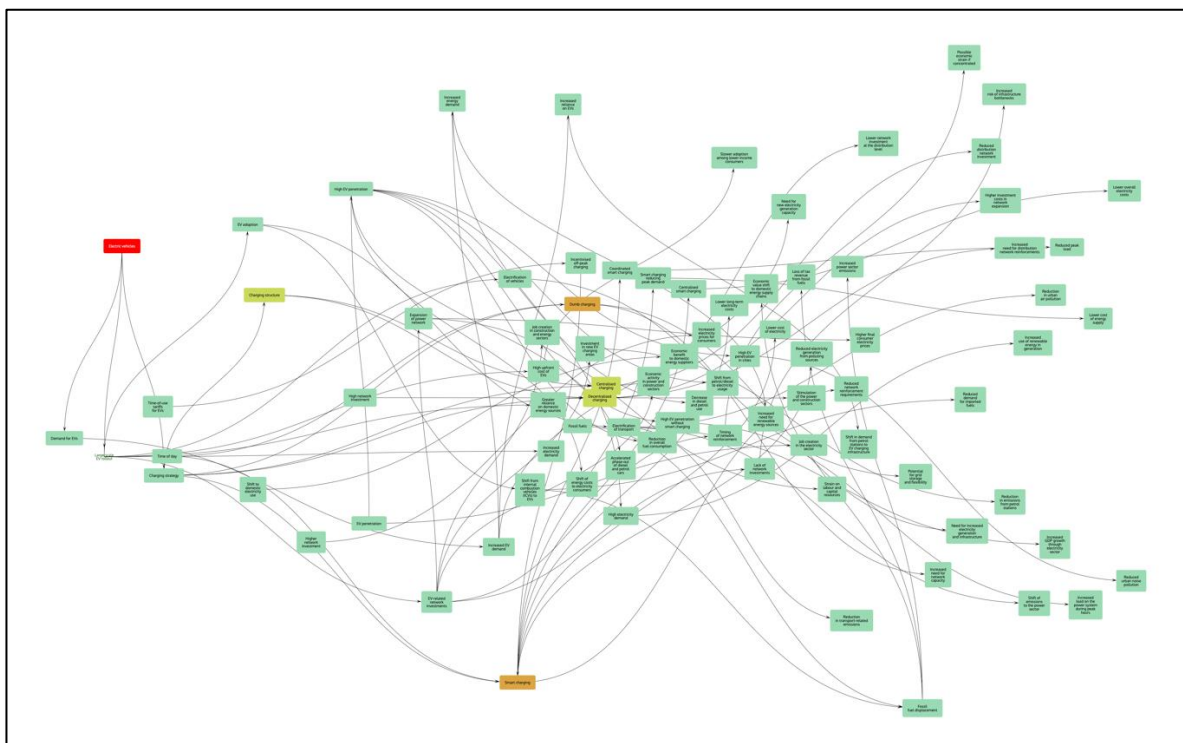


Figure 9: Example system map on electric vehicle adoption in the UK visualised in PRSM using cause-and-effect data extracted from Calvillo and Turner (2020) using ChatGPT.

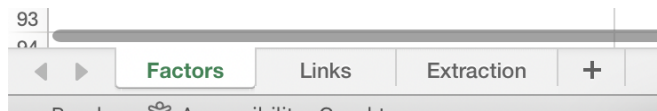
²⁰ This map contains 85 nodes (factors) and 83 edges (links between factors). ChatGPT originally generated 109 factor labels from the source document. 24 of these factors were manually removed: 5 of these were erroneous, while the remaining 19 were duplicates.

Health warning: please keep in mind the following guidance explains how to turn the cause-and-effect pairs generated by the LLM into a fully visualised system map. However, it does not provide guidance on exactly how to clean, refine, and add to the causal links identified by the LLM, nor do we offer detailed guidance on how to visualise the map. As stated at the start of this document, you should treat the map from the LLM as preliminary. You should use your own judgement on what to do next with the map, considering your project purpose. We envisage these stages of refining the map, merging it with other work, and visualising it, as just as important, if not more so, than the generation of the map with an LLM.

Formatting the exported list

The following guidance refers to the Excel workbook outlined above containing the list of cause-and-effect pairs generated by the LLM.

- Open the Excel workbook containing the cause-and-effect pairs.
- Go to *File* → *Save a copy*.
- Save the file using a new name. We recommend including the policy document name and *PRSM* in the file name.
- Create two new sheets; named **Factors**, and **Links**, respectively. Then, change the name of the sheet containing the cause-and-effect pairs to **Raw Data**.



As we explain below, the *Factors* and *Links* sheets need to be formatted in a specific way to be interpreted by PRSM. PRSM simultaneously refers to both *causes* and *effects* as “*labels*”, i.e. the labels which represent individual system factors. The *Factors* list is a list of all unique labels, while the *Links* list represents each causal relationship between those factors.

Creating the **Factors** list

- The *Factors* and *Links* sheets need to be formatted in a specific way to be interpreted by PRSM. In the *Factors* sheet, enter the following text into cells A1:C1 to create column names.

	A	B	C
1	Label	Description	Style
2			

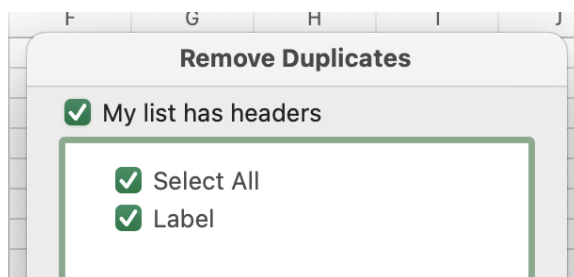
- Navigate to the *Raw Data* sheet.

- Highlight all of the factor names contained within the Causes column, taking care to not include the column name (Causes).
- Copy the highlighted cells.
- Navigate back to the *Factors* sheet.
- Paste the text from the copied cells into cell A2 so they appear in the *Label* column.
- Navigate back to the *Raw Data* sheet.
- Highlight all of the factor names contained within the Effects column, taking care to not include the column name (Effects).
- Navigate back to the *Factors* sheet.
- Paste the text from the copied cells into the next blank cell in column A, beneath the factors you previously pasted into the column.

Removing duplicate factors

To avoid factors appearing multiple times in the map, you will need to remove duplicates from the *Factors* sheet.

- Navigate to the *Factors* sheet.
- Select column A so the whole column is highlighted.
- Go to *Data* → *Table Tools* → *Remove Duplicates*
- A dialog box will appear. Choose **Continue with the current selection** and then click **Remove Duplicates...**
- A second dialog box will appear. Ensure **My list has headers** and **Select All** are selected, along with the column name **Label** as below. Then click OK.



- Any duplicate factor names will be removed from the list of factors, ensuring they only appear once in the map.

Creating the *Links* list

- The *Factors* and *Links* sheets need to be formatted in a specific way to be interpreted by PRSM. In the *Links* sheet, enter the following text into cells A1:D1 to create column names.

	A	B	C	D
1	From	To	Description	Style
2				

- Navigate to the *Raw Data* sheet.
- Highlight all of the factor names contained within both the *Causes* and *Effects* columns, taking care to not include the column names.
- Copy the highlighted cells.
- Navigate back to the *Links* sheet.
- Paste the text from the copied cells into cell A2 so they appear under the *From* and *To* columns.

Adding additional factors

If you have additional factors relevant to this policy system that you wish to include in the map, you can add these to the *Factors* and *Links* sheets at this stage. You must ensure that you use the exact same labels in both sheets for these to be displayed in the map.

Deleting the raw data

The raw data needs to be removed, as this is not readable by PRSM.

- After following the above steps, delete the *Raw Data* sheet.
- Save the file.

Formatting the map

PRSM uses the *Description* and *Style* columns to respectively categorise the factors and format their appearance.

To apply basic formatting to the systems map, add the following description and style for each entry in both the *Factors* and *Links* sheets:

- Description: *factor*
- Style: *1*

These fields can be customised to create maps with factors formatted using varying colours and shapes. Detailed guidance on how to apply descriptions and styles is available on the PRSM website.

Can this be automated using an LLM?

We tried prompting ChatGPT to format the tabular list of cause-and-effect pairs into an Excel spreadsheet suitable for use in PRSM, but it failed to do so. Therefore, we recommend doing this manually.

Importing the list into PRSM

To create a basic visualisation of your policy system in PRSM:

- Navigate to the PRSM website and click *Start Now*.
- PRSM will open. In the top ribbon bar, click *Open*.
- Open the Excel workbook that you formatted for PRSM above.
- PRSM will create a visualisation of the system. At this point, the factors may appear jumbled up and illegible.
- In the top ribbon bar, click *Settings*.
- A sidebar will appear. In this sidebar:
 - Under *Analysis*, you will find *Size Factors To*.
 - Select *Equal*.
 - Under *Network*, you will find *Auto Layout*.
 - Select a layout. We recommend starting with *Barnes Hut*.
 - The factors will become equally sized and shift outward, making the visualisation more legible. If the factors are clustered too close together, select *Barnes Hut* again until they are legible.

Method and Limitations

To develop this guidance, we used the following materials and methods.

Materials

We conducted a review of literature spanning LLMs, extraction of causal statements and prompt engineering. We summarised a list of techniques which formed the structure of our testing.

A sample set of four academic papers (Papers 1-4) discussing decarbonisation policy in the UK and China was used as the source data from which the LLM would extract causal statements.

A further three documents (Papers A-C) were used to test the validity and broad applicability of our guidance beyond decarbonisation policy. To reduce the likelihood that they were already part of the training data of the LLM, we restricted our selection of these three documents to those which had been published in or after September 2024.

Paper	Name	Author(s)	Type
1 [†]	Analysing the impacts of a large-scale EV rollout in the UK – How can we better inform environmental and climate policy? [UK]	Calvillo and Turner (2020) [link]	Academic Paper
2	Energy transitions and uncertainty: Creating low carbon investment opportunities in the UK electricity sector [UK]	Bolton, Foxon and Hall (2016) [link]	Academic Paper
3	Rapid decarbonization in the Chinese electric power sector and air pollution reduction Co-benefits in the Post-COP26 Era [China]	Jiang (2023) [link]	Academic Paper
4	The water-carbon constraints' impact on the development of coal power industry in the Yellow River Basin [China]	Jiefang and Chenmeng (2022) [link]	Academic Paper
A	Restorative justice: Adopting a whole system approach to address cultural barriers in criminal justice	Banwell-Moore (2024)	Academic Paper

B	Discretionary Housing Payments	Rachael Harker (House of Commons Library, UK Parliament)	Research Briefing
C	Design for Life Roadmap: Building a circular economy for medical technology	Department of Health & Social Care (UK Government)	Policy Paper

For tests with multiple documents, all four academic papers were uploaded alongside the prompt. ¹For tests with a single paper, Paper 1 was uploaded alongside the prompt to control for variation in document content and allow for comparison between generated outputs. Consequently, since uploading multiple documents resulted in weaker performance, Paper 1 became our *de facto* primary testing document.

Our testing focused predominantly on GPT-4o, which at the time of writing was the highest-quality *closed* but commercially-available LLM, according to *ArtificialAnalysis.ai*.

Method

Each technique was added to a list of tests which were ordered by complexity, with simpler techniques tested first and more complex techniques (e.g. one-shot/few-shot prompting) tested later.

Starting with a simple prompt, we changed the prompt text to incorporate a given technique. The updated prompt was submitted to ChatGPT, and the generated output was inspected, to see whether an improvement was made. Each technique was tested three times; an initial attempt was made, followed by a *regeneration* of the initial output in the same window, and the prompt submitted in a separate chat window.

The output prompt a given prompt and test was examined using number of cause-and-effect pairs identified as a metric which could be compared across tests to give a measure of reliability, along with an inspection of the list to see whether the pairs generated appeared accurate.

✅ If the output (as a list of cause-and-effect pairs) improved, the technique was deemed suitable for this task, and we accepted the new prompt. This new prompt would be passed through to the next test. Each of the previously listed techniques were deemed suitable as their tests were successful.

❌ If the output worsened, the technique was deemed unsuitable, and we rejected the new prompt. The original prompt would be passed through to the next test.

Therefore, our *example prompt* was *engineered* iteratively and incorporates all of the techniques which resulted in an observed improvement in the quality of the output generated by the LLM.

Limitations

Out-of-the-box guidance

The use of AI tools such as LLMs in systems thinking and causal inference is a nascent topic. Yet over the last couple of years, AI tools have become more powerful, leading to rapid adoption by individuals and organisations. Consequently, we suspect that tools such as LLMs are already becoming part of the modern policymaker's workflow.

For this reason, we have focused on providing step-by-step guidance to scaffold the use of an LLM that policymakers are likely to be familiar with already. Our guidance can be used "out of the box", i.e. used immediately. This is in lieu of an extensive examination of the different tools, methods and strategies that could be used to generate systems maps from policy documents.

Playground

Our testing focused on using the basic web interface of ChatGPT. Some readers may be aware that a premium subscription provides access to the *OpenAI Playground*, which provides access to advanced customisation features. To ensure our guidance remained relevant to our intended audience of policy analysts and researchers, we decided to restrict our testing to the version of ChatGPT that is widely accessible. Therefore, our testing did not verify whether output quality could be improved by varying model configuration parameters or using custom "assistants".

Relationships between factors

As an output of our testing, this guidance solely focuses on using LLMs to identify and extract causal statements from texts in the form of pairs of causes and effects, which form the nodes and edges of a system map. There remain a number of aspects of relationships between system factors that we did not address in our testing or guidance. We present these as areas for future research.

Our testing did not establish whether LLMs can:

- Identify and distinguish between associative, correlative and causal relationships.
- Explicitly identify direction of causality within causal statements. ChatGPT was able to define this implicitly, as by definition a cause-and-effect pair has clear direction of causality: from *cause* to *effect*. However, there could have been cases of bidirectionality in our sample papers, where A causes B, but B also causes A independently, which were unaccounted for. Earlier versions of

ChatGPT may inaccurately identify the direction of causality from policy documents²¹.

- Identify feedback loops within the document and/or generated list and whether these are balancing or reinforcing loops.

Iterative method

As outlined above, we used an iterative method to test each of the techniques that we had identified in succession. This strategy allowed us to test a wide range of techniques at the expense of comprehensiveness. Testing each technique in succession restricted the combination of techniques that were used. However, testing every possible permutation of the list of techniques and combination of techniques would have been extremely time intensive. As a result, there may be other possible permutations to the ordering of our techniques that could have produced a final prompt which leads to a more accurate output being generated. We encourage researchers and analysts to test these techniques themselves and welcome further research which builds our findings.

Multiple shot prompting

As discussed previously, we found that including an example cause-and-effect pair in the prompt (known as “one-shot” prompting) did not result in an increase in the quality of the output list generated. Therefore, we did not test including a larger number of examples (known as “few-shot” prompting).

Similarly, policy researchers and analysts may already have a documented list of relevant system factors which they wish to combine with the LLM-generated cause-and-effect pairs. Our testing did not verify whether these can be successfully incorporated into the list by the LLM, particularly as already known factors may not be causally related. We present this as an area for future research, where novel approaches such as chain-of-thought (CoT) prompting and self-consistency decoding may offer further optimisation of the prompt.

Memory feature & newer models

During our testing, OpenAI released a *memory* feature which retains knowledge from historical chats. To reduce bias during our testing, we opted out of this feature. More generally, it remains unclear whether long-term memory of chats has any impact on the quality of subsequent outputs generated for the purpose of systems mapping.

²¹ Giabbanelli and Witkowicz (2024) show that while GPT-3.5 accurately identified nodes and edges from causal statements within a sample of policy documents, its identification of the direction of causality was largely inaccurate.

Our testing also began before the release of the o-series models, GPT-4.5 model and *Deep Research* tool. Therefore, our guidance does not account for any differences in output which are realised when using these new models instead.

Authors

Jordan White is a master's student of Economics with Innovation at the University of Bristol. He is a Research Assistant at the Institute for New Economic Thinking (INET) at the University of Oxford and has previously worked for Ministry of Justice Digital & Technology. He is interested in using computational methods, behavioural science, participatory design and futures studies as tools to identify and evaluate the dynamic role of technology in society.

Dr. Pete Barbrook-Johnson is a Departmental Research Lecturer at the University of Oxford. He is a social scientist, economist, complexity scientist, and systems thinker. He regularly uses research methods such as agent-based modelling and systems mapping in his applied environmental, energy, and public health research and policy analysis. He teaches on a range of undergraduate and master's courses, focussing on the economics of environmental change, and the use of complexity and systems sciences in environmental issues.

Much of the research upon which this guidance document is based was conducted as part of the *UNIQ+* internship programme at the University of Oxford during July and August 2024.

Contact information

Jordan White jordan.white@bristol.ac.uk

Dr. Pete Barbrook-Johnson peter.barbrook-johnson@ouce.ox.ac.uk

Resources

Below, we highlight useful resources for further reading on systems mapping and prompt engineering.

Systems Mapping *How to build and use causal models of systems*

Pete Barbrook-Johnson, Alexandra S. Penn

Palgrave Macmillan

eBook ISBN: 978-3-031-01919-7

<https://doi.org/10.1007/978-3-031-01919-7>

Prompt Engineering for Generative AI *Future-Proof Inputs for Reliable AI Outputs*

James Phoenix, Mike Taylor

O'Reilly Media, Inc

ISBN: 9781098153434

<https://www.oreilly.com/library/view/prompt-engineering-for/9781098153427/>

Beginner's Guide to OpenAI API *Build your own LLM tool from scratch*

Chanin Nantasenamat

<https://medium.com/data-professor/beginners-guide-to-openai-api-a0420bc58ee5>

Ethical Costs and Epistemic Risks of LLMs: A Resource for Psychology Researchers

M.J. Crockett, Department of Psychology, Princeton University

<https://docs.google.com/document/d/1Fv9TDqghKOQAMUyLNdxUI-eaQIWc2QAuTx0bSJsc-yU/edit?usp=sharing>

Participatory Systems Mapping: a practical guide

Alexandra S. Penn, Pete Barbrook-Johnson

Centre for the Evaluation of Complexity Across the Nexus (CECAN), University of Surrey

<https://www.cecan.ac.uk/wp-content/uploads/2020/09/PSM-Workshop-method.pdf>

References

Artificial Analysis, 2024. *LLM Leaderboard - Comparison of GPT-4o, Llama 3, Mistral, Gemini and over 30 models*. [online] Available at: <https://artificialanalysis.ai/leaderboards/models>. [Accessed 20 December 2024]

Bender, E.M., Gebru, T., McMillan-Major, A. and Shmitchell, S., 2021. *On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?* 🦜 [online] FAccT '21, March 3–10, 2021, Virtual Event, Canada (pp.616-617). Available at: <https://dl.acm.org/doi/pdf/10.1145/3442188.3445922>

Calvillo, C.F. and Turner, K., (2020). Analysing the impacts of a large-scale EV rollout in the UK – How can we better inform environmental and climate policy? <https://doi.org/10.1016/j.esr.2020.100497>

Centre for Research in Social Simulation [CRESS] (2024). PRSM: Participatory Systems Mapper Homepage. Available at: <https://prsm.uk/index.html>

Giabbanelli, P. and Witkowicz, N., 2024. *Generative AI for Systems Thinking: Can a GPT Question-Answering System Turn Text into the Causal Maps Produced by Human Readers?* [online] Proceedings of the 57th Hawaii International Conference on System Sciences | 2024 (p.7547). 978-0-9981331-7-1. <https://hdl.handle.net/10125/107291>

Macri, A. and Spong, G., 2023. Crafting prompt sandwiches for generative AI. [online] Available at: <https://www.elastic.co/blog/crafting-prompt-sandwiches-generative-ai> [Accessed 02 March 2025].

Mizrahi, G. and Serfaty, D., 2024. *Unlocking the Secrets of Prompt Engineering: Master the Art of Creative Language Generation to Accelerate Your Journey from Novice to Pro*. [book]. First edition. Birmingham, England: Packt Publishing.

Powell and Caldas Cabral., 2023. Using AI to turn stakeholders' stories into causal maps, rapidly and at scale. Causal Map Ltd. EES Online Conference 2023.

Pudjianto et al., 2013. Smart control for minimizing distribution network reinforcement cost due to electrification, *Energy Pol.* 52 (Jan. 2013) 76–84. Available at: <https://doi.org/10.1016/j.enpol.2012.05.021>

Spyscape, 2024. *The Ultimate Guide to Engineering Effective ChatGPT Prompts*. [online] Available at: <https://spyscape.com/article/chatgpt-prompt-engineering>. [Accessed 02 March 2025]

Technical Life, 2024. *How to expert ChatGPT prompt*. [online] Available at: <https://medium.com/@luketong/how-to-expert-chatgpt-prompt-4a3e26233597>. [Accessed 02 March 2025]

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai.
2016. Man is to computer programmer as woman is to homemaker? debiasing word
embeddings. In Advances in Neural Information Processing Systems. 4349–4357.
<https://doi.org/10.1145/3351095.3372843>

Appendices

Appendix A: Ranking of highest quality LLMs as of December 2024.

Table A1 lists the top ten proprietary LLM models in rank order by quality index according to an independent ranking by *ArtificialAnalysis.ai* in December 2024.

Rank (by Quality Index)	Model	Creator (Developer)	Quality Index (Normalised Average)
1	o1-Preview	OpenAI	86
2	o1-mini	OpenAI	84
3	Gemini 2.0 Flash (exp)	Google	82
4	Gemini 1.5 Pro (Sep)	Google	81
5	Claude 3.5 Sonnet (Oct)	Anthropic	80
6	GPT-4o (Aug '24)	OpenAI	78
7	GPT-4o (May '24)	OpenAI	78
8	Qwen2.5 72B	Alibaba	77
9	Claude 3.5 Sonnet (June)	Anthropic	76
10	GPT-4 Turbo	OpenAI	75

Table A1: Ranking of proprietary LLMs by quality index as of December 2024. Data adapted from Artificial Analysis (2024).

Appendix B: Generating system map visualisations natively within ChatGPT.

This section refers to the discussion under [Unsuccessful techniques](#) (Page 33).

Basic prompt

The following images demonstrate the output from ChatGPT in response to a basic prompt for a system map to be generated.

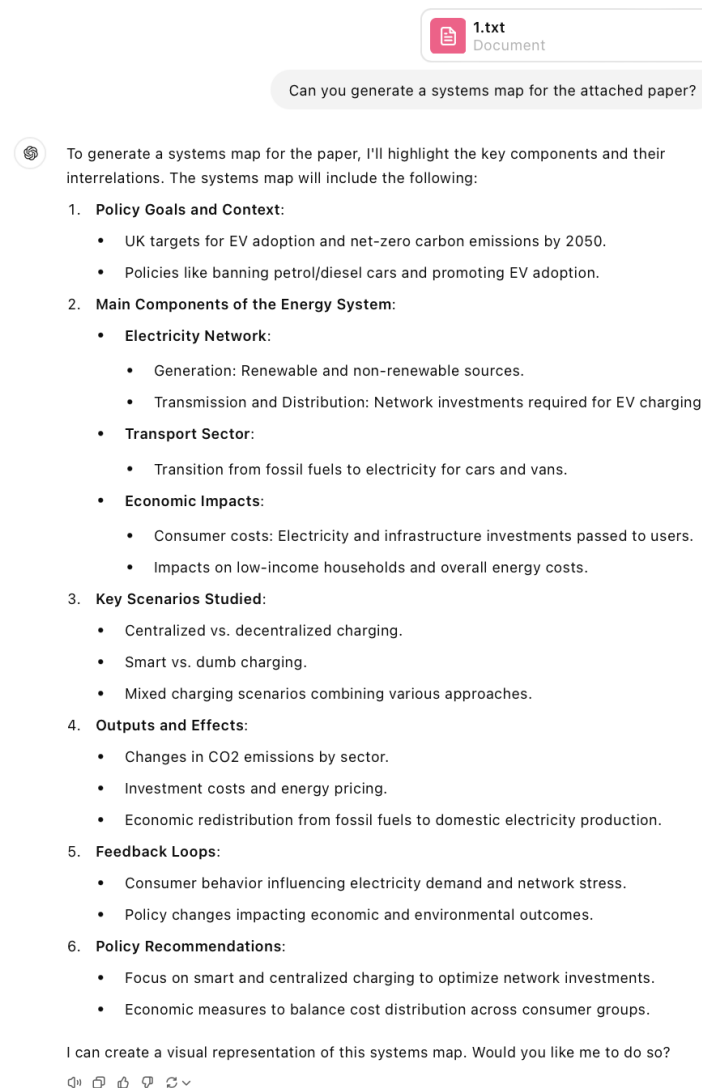
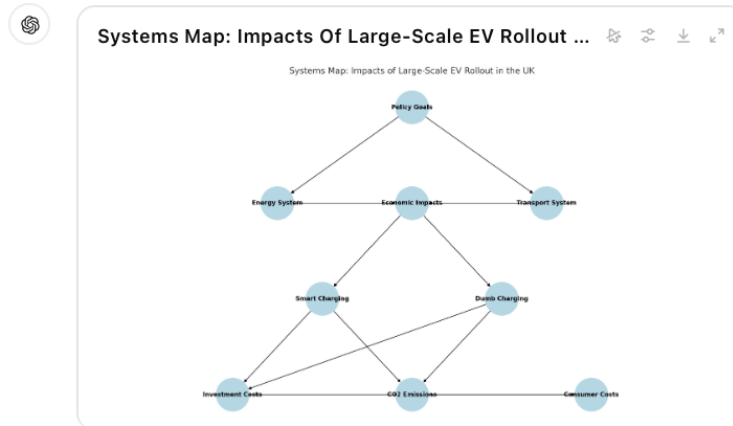


Figure 10: Initial response from ChatGPT.

I can create a visual representation of this systems map. Would you like me to do so?

Yes



Here is the systems map visualizing the key components and their interrelations for the impacts of a large-scale EV rollout in the UK. Let me know if you need further adjustments or additional details! [\[↗\]](#)

🔊 📄 👍 🗣️ ↺ ✓

Figure 11: System map generated by ChatGPT from a simple prompt.

```

1 import matplotlib.pyplot as plt
2 import networkx as nx
3
4 # Initialize the graph
5 G = nx.DiGraph()
6
7 # Define nodes and their attributes
8 nodes = {
9     "Policy Goals": {"pos": (0, 6)},
10    "Energy System": {"pos": (-3, 3)},
11    "Transport System": {"pos": (3, 3)},
12    "Economic Impacts": {"pos": (0, 3)},
13    "Smart Charging": {"pos": (-2, 0)},
14    "Dumb Charging": {"pos": (2, 0)},
15    "Investment Costs": {"pos": (-4, -3)},
16    "CO2 Emissions": {"pos": (0, -3)},
17    "Consumer Costs": {"pos": (4, -3)},
18 }
19
20 # Add nodes to the graph
21 for node, attr in nodes.items():
22     G.add_node(node, **attr)
23
24 # Define edges and their relationships
25 edges = [
26     ("Policy Goals", "Energy System"),
27     ("Policy Goals", "Transport System"),
28     ("Energy System", "Economic Impacts"),
29     ("Transport System", "Economic Impacts"),
30     ("Economic Impacts", "Smart Charging"),
31     ("Economic Impacts", "Dumb Charging"),
32     ("Smart Charging", "Investment Costs"),
33     ("Smart Charging", "CO2 Emissions"),
34     ("Dumb Charging", "Investment Costs"),
35     ("Dumb Charging", "CO2 Emissions"),
36     ("Investment Costs", "Consumer Costs"),
37     ("CO2 Emissions", "Consumer Costs"),
38 ]
39
40 # Add edges to the graph
41 G.add_edges_from(edges)
42
43 # Get node positions for visualization
44 pos = {node: attr["pos"] for node, attr in nodes.items()}
45
46 # Draw the graph
47 plt.figure(figsize=(10, 8))
48 nx.draw(
49     G, pos, with_labels=True, node_size=3000, node_color="lightblue", font_size=10, font_weight="bold"
50 )
51 plt.title("Systems Map: Impacts of Large-Scale EV Rollout in the UK", fontsize=14)
52 plt.show()

```

Figure 12: Python code used by ChatGPT to generate the visualisation in Figure B2²².

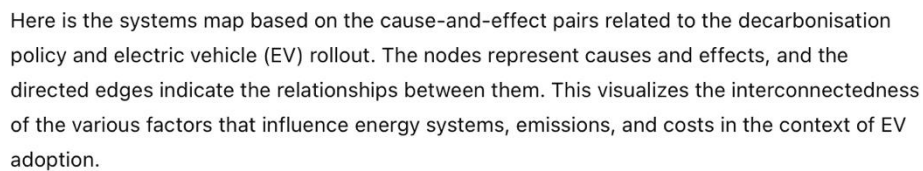
Example prompt

The following images demonstrates the output from ChatGPT in response to a subsequent prompt for a system map to be generated, after using both stages of our example prompt (Page 31).

Can you generate a systems map using this table?

Figure 13: Third-stage prompt used after the first two stages of our ‘example prompt’.

²² The script was copied from ChatGPT into Visual Studio Basic to conveniently display the entire script in one image.



If you need any further modifications or additional visualizations, feel free to ask! [\[>\]](#)



Figure 14: System map generated natively within ChatGPT, using the third-stage prompt (Figure B4) and cause-and-effect pairs generated using Paper 1.