AI & SOCIETY
https://doi.org/10.1007/s00146-025-02487-4

REVIEW

# Leveraging large language models for thematic analysis: a case study in the charity sector

**Chuanchi Wen[1] · Paul Clough[1,2] · Rachel Paton[3] · Rebecca Middleton[3]**

Received: 13 March 2025 / Accepted: 3 July 2025


**Abstract**
This study explores how large language models (LLMs) can support deductive and inductive thematic coding in real-life contexts, balancing AI-driven efficiency with essential human oversight. Using three datasets from Tearfund, a UK-based Christian charity, we propose a dual-role human–LLM collaborative framework where the LLM functions as an initial annotator and a validator. In the deductive phase, GPT-4o and GPT-4o-mini were compared against human coders. GPT-4o achieved a substantial agreement in multi-label thematic categorization ($\kappa = 0.61$–$0.65$), while GPT-4o-mini showed a moderate agreement ($\kappa = 0.41$–$0.58$). Both models excelled in sentiment analysis ($\kappa = 0.91$–$0.95$), but struggled with evaluating evidence of impact due to contextual complexity ($\kappa \leq 0.01$). GPT-4o-mini exhibited greater output variability and instability than GPT-4o, but benefited more from few-shot learning to mitigate hallucinations. In the inductive phase, GPT-4o demonstrated a strong semantic alignment with human-generated themes (cosine similarity = 0.76–0.79) though its tendency toward broad themes required human refinement. Despite their potential to streamline thematic analysis, LLMs also pose limitations and implementation challenges, including inconsistencies in excerpt extraction (precision = 0.41, recall = 0.53) and the trade-off between the time saved in coding and the time required for human validation. To facilitate practical implementation, we provide reusable prompt templates for four stages: context, instructions, data processing, and verification. Our findings underline the indispensable role of human expertise—from prompt engineering and managing hallucinations to final verification—to ensure accurate and trustworthy AI-assisted analyses. While LLMs can enhance qualitative analysis, their full potential is only realized under skilled human guidance.

**Keywords** Large language models (LLMs) · Generative AI (GenAI) · GPT-4o · Prompt engineering · Thematic analysis

## 1 Introduction

Recent advances in AI, particularly Generative Artificial Intelligence (GenAI) such as large language models (LLMs), have greatly expanded the applications of AI in various fields (Sedkaoui & Benaichouba 2024), including the charity sector. For example, the 2023 Charity Digital Skills report [1]suggested that 35% of charities were already using AI for tasks, such as writing fundraising materials or taking meeting minutes, with a further 26% having plans to do so in future. This growing trend is evident in Tearfund, a UK-based Christian charity working globally to tackle poverty and injustice as it explores how AI-driven thematic analysis can assist in evaluating projects and informing future initiatives. Unlike traditional natural language processing (NLP) techniques, LLMs provide an intuitive, text-based interface, making them accessible to a broader user base without deep programming knowledge (Turobov et al. 2024). This ease of use has driven the interest in applying LLMs to thematic analysis, a widely used method in qualitative research to

✉ Paul Clough
p.d.clough@sheffield.ac.uk

Chuanchi Wen
chuanchiwen@gmail.com

Rachel Paton
rachel.paton@tearfund.org

Rebecca Middleton
rebecca.middleton@tearfund.org

[1] University of Sheffield, Sheffield, United Kingdom

[2] TPXimpact, London, United Kingdom

[3] Tearfund, London, United Kingdom

[1] Charity Digital Skills Report (site visited: 27/11/2024): https://charitydigitalskills.co.uk/wp-content/uploads/2023/07/Charity-Digital-Skills-Report-2023.pdf

Published online: 17 August 2025

systematically uncover patterns or themes (Terry et al. 2017).

Carrying out thematic analysis enables researchers to explore research questions or gain insights into a specific issue by identifying and examining themes (Braun & Clarke 2006; Maguire & Delahunt 2017). Thematic analysis can involve both deductive and inductive coding approaches. Deductive coding, also known as theory-driven coding, applies an existing framework or predefined codes to the data, often based on specific theories or hypotheses the researchers wish to explore. In this method, researchers use a detailed "codebook," developed from an established theoretical framework, to guide the coding process and ensure that the data accurately reflect the predefined concepts or the characteristics being examined (Miles et al. 2014). In contrast, inductive coding, or data-driven coding, generates codes and themes directly from the data itself, allowing them to emerge naturally without the influence of preconceived categories (Thomas 2006). This approach is particularly useful in exploratory research, aiming to discover new insights (Mansourian 2008).

Although effective, thematic analysis is often time-consuming, labor-intensive and subjective (Dai et al. 2023; Gamieldien et al. 2023; Tai et al. 2024). Integrating LLMs into thematic analysis could offer several benefits, including reduced labor costs (Gilardi et al. 2023), increased efficiency (Gao et al. 2024; Zhang et al. 2024), adding an additional layer of objectivity (Mathis et al. 2024), and the ability to uncover insights that human coders might overlook (Meng et al. 2024; Turobov et al. 2024). However, the effective application of LLMs in thematic analysis depends on a successful prompt design to guide the models and mitigate the risk of generating "hallucinations"—inaccurate or misleading information that can compromise analysis reliability (Tai et al. 2024; Turobov et al. 2024). Although LLMs are gaining popularity for thematic analysis, their practical application remains in early stages. Most studies focus on either deductive coding or inductive coding, with some employing them solely as initial coders and a few extending their role to include validation. However, research on combining both coding approaches and dual roles into a comprehensive framework is limited. Additionally, much of the existing work does not consider the use of AI technologies within the context of real-life coding workflows.

This study addresses these gaps through a case study with Tearfund. We investigate the role of LLMs, particularly GPT-4o, in supporting Tearfund's deductive and inductive coding workflows, which form the analysis phase of an evaluation meta-synthesis (explained further in Sect. 3.1). Specifically, we address the following research questions: (RQ1): How well can LLMs perform deductive and inductive coding tasks? (RQ2): How can LLMs be implemented within an existing thematic analysis workflow? Although centered around a specific use case, we believe the practical insights gained from this investigation will be of interest to any organization seeking to employ AI assistance for thematic analysis. In addition to evaluating the performance of LLMs, we also propose a dual-role human–LLM collaborative framework where LLMs act as both validators and initial annotators under human oversight. Our findings support the efficacy of LLMs for thematic analysis as demonstrated in past work, but also highlight issues with their practical implementation in real-life contexts.

## 2 Literature review

### 2.1 Large language models (LLMs)

LLMs have driven notable progress in NLP (Patil & Gudivada 2024), offering tools that can significantly enhance traditional qualitative analysis methods, such as thematic analysis. By automating text analysis and generating codes, LLMs showcase the potential to complement human expertise in identifying themes and patterns in large datasets (Khan et al. 2024). Several cutting-edge LLMs have emerged, including GPT-4, GPT-4o, PaLM2, Llama 3.1, and Claude 3.5 Sonnet, from companies, such as OpenAI, Meta, Google, and Anthropic. These foundational models are pre-trained with billions to trillions of parameters, enhancing accuracy and coherence in language tasks, and can be used for multiple downstream tasks. Additionally, their expanded context windows, such as the 128,000-token capacity of GPT-4o and the 200,000-token capacity of Claude 3.5 Sonnet, allow them to handle vast amounts of input data efficiently and be used for more complex tasks.

The success of using LLMs, particularly in thematic analysis, hinges on the quality and specificity of the prompts provided (De Paoli 2023; Zhang et al. 2025). A well-crafted prompt, which can be a question, command, or statement, is used to guide the model's focus (Giray 2023). Techniques, such as in-context learning (ICL), chain-of-thought (CoT) prompting, and role-playing, can improve model outputs by providing examples (one- or few-shot learning), breaking reasoning into steps, and aligning outputs with specific domains (Brown et al. 2020; Wei et al. 2022; Gao 2023). Clearly structuring prompts, particularly for complex tasks, can also enhance the model's response quality (Arvidsson & Axell 2023). Designing an effective prompt is a clear opportunity for human engagement, especially for thematic analysis that often requires domain and subject expertise (Turobov et al. 2024; Zhang et al. 2025) and can be significantly improved with iterative refinement (Bijker et al. 2024; De Paoli 2023; Hou et al. 2024; Khan et al. 2024; Sinha et al. 2024).

Despite the benefits of LLMs, they also pose a number of challenges. For example, LLMs inherently carry the biases in their training data which can result in prejudiced and unfair outputs (Motoki et al. 2024). The relative ease with which LLMs can be used for tasks, such as thematic analysis, can also result in an unhealthy trust in the quality of outputs resulting in a lack of critical analysis and engagement with the process (Lee et al. 2025). There are also a range of other ethical issues raised by LLMs, such as their impact on human workers and the potential for job displacement, inequalities in the access of AI technologies, the environmental costs of the vast computing infrastructure needed to power applications like ChatGPT, the potential for dissemination of misinformation, and the lack of transparency within "black box" systems in their decision-making and internal reasoning processes (Hendrycks 2024; Shin 2025). Additionally, LLMs can generate inaccurate or false information, a phenomenon known as "hallucination" (Do et al. 2024; Turobov et al. 2024). This arises when models struggle with conflicting or diverse data. In the case of thematic analysis, human validation of LLM outputs ensures accuracy and completeness (Lee et al. 2024). Mitigating hallucinations can involve prompt refining and self-assessment strategies (Cooke 2024) and is vital in practice for gaining users' trust. In this study, we use various prompt techniques to guide and improve the model outputs.

### 2.2 LLMs for thematic analysis

As LLMs evolved, researchers have explored their potential application in thematic analysis. For example, Xiao et al. (2023) investigated the use of GPT-3 for deductive coding, demonstrating GPT-3 can be a useful tool for deductive qualitative coding. Their findings showed that when properly guided with a codebook, GPT-3 reduced the labor-intensive nature of manual coding while maintaining a fair to substantial level of agreement with expert coders. They also demonstrated that the codebook-centered design outperforms the example-centered designs and highlighted examples that are crucial to performance, especially when moving from a zero-shot to a one-shot setting. In a different application of deductive coding, Tai et al. (2024) employed ChatGPT-3.5 to execute predefined codes over 160 iterations. They introduced a new approach called LLMq to measure how consistently the LLM identified the relevant codes. The finding revealed that consistency was achieved after several iterations, demonstrating the potential of LLMs, not only as supplementary tools, but also as means to uncover previously overlooked insights.

In contrast, other studies have focused on the use of LLMs for inductive coding. Gao et al. (2024) developed CollabCoder, a workflow implemented on web-based platform integrating GPT-3.5-turbo to support the inductive qualitative analysis process. This platform facilitated various stages of the qualitative analysis process, including code suggestions, team discussions, and codebook development, enabling collaborative theme generation. De Paoli (2023) applied Braun and Clarke's (2006) six-phase framework to evaluate GPT-3.5-Turbo for inductive thematic analysis of semi-structured interviews. While the model successfully inferred many key themes, it missed some important ones identified by human analysts, highlighting the importance of a human-in-the-loop approach and the necessity of human involvement to ensure comprehensive and valid analysis (Khalid & Witmer 2025).

Dai et al. (2023) adopted a more comprehensive approach by leveraging GPT-3.5-Turbo-16 k for both deductive and inductive coding, further broadening the scope of using LLMs for thematic analysis. Their work introduced the LLM-in-the-loop model, which showcased a collaborative approach where human experts and LLM worked together iteratively throughout the coding process. This model achieved a near-perfect agreement between a human and machine coders, demonstrating that LLMs can significantly reduce labor and time while maintaining results comparable to those produced by human coders.

Building on this fact, several studies have explicitly proposed human–LLM collaborative frameworks. Yan et al. (2024) highlighted an effective human–AI collaboration framework, emphasizing the supportive role of LLMs in assisting human coders, rather than replacing them. Additionally, Meng et al. (2024) presented CHALET, a novel methodology for qualitative research that combines human and LLM efforts. CHALET integrates LLMs to assist with data collection and employ a dual coding approach: both human and LLM perform deductive coding to pinpoint disagreements, which are then collaboratively analyzed through inductive coding to uncover new insights.

Other research, though not directly addressing a collaborative framework, still centers on the concept of human–LLM cooperation. Earlier studies suggested using LLMs as initial coders to improve efficiency (Chew et al. 2023; Turobov et al. 2024). De Paoli (2023), for instance, explored the use of GPT-3.5-Turbo as an initial coder in inductive coding, but recommended its future use as a validation tool. Recent studies have expanded LLMs' roles as both initial coders and 'second coders' to validate human coding, thereby enhancing overall coding quality by uncovering patterns human coders may neglect (Bijker et al. 2024; Tai et al. 2024; Zhang et al. 2024). Additionally, LLMs have also been suggested as mediators to reconcile differences between human coders (Gao et al. 2024), further reinforcing the idea that LLMs work best in conjunction with human expertise rather than as a replacement for human coders. In this research, we explore the value of LLMs in thematic analysis within a dual-role human–LLM collaboration framework.

## 3 Methodology

### 3.1 Description of the Case Study

Tearfund[2] is a UK-based Christian charity working globally to implement practical and sustainable programs that address poverty and injustice. In this work, we ground the use of LLMs to support qualitative analysis within Tearfund as a case study, an approach enabling "an in-depth appreciation of an issue, event or phenomenon of interest, in its natural real-life context" (Crowe et al. 2011:1). This allowed us to study the context in which LLMs could be used, generate realistic test data, and gather feedback from staff regarding their utility.

Tearfund regularly evaluates the projects they and their partners support or deliver, ensuring wise use of resources and deriving valuable lessons for future endeavors. In most cases, the output of such an evaluation is a comprehensive evaluation report, in which the evaluator analyzes primary and secondary data; draws conclusions about the effectiveness, efficiency, relevance, impact, and sustainability of the project; and makes specific recommendations for future work. Since 2019, on an annual basis, Tearfund's Impact team has collated these evaluation reports and conducted further analysis on them; an exercise referred to as an evaluation meta-synthesis. The purpose of the meta-synthesis is to promote organizational learning, by summarizing evaluation findings, making them digestible and accessible to a wider internal audience, and highlighting findings that are potentially of strategic importance.

The exercise starts with a human coder reading each evaluation report in full and scoring it against the Bond Evidence Principles,[3] which are widely used in the sector to measure evidence quality. Reports scoring 30 or more are included in the subsequent stages of the analysis, which employ both deductive and inductive coding methods (either being performed first). At this preliminary stage, Tearfund staff also group the evaluation reports according to the organization's four corporate priorities: CCT (Church and Community Transformation), RPS (Reconciled, Peace-filled Societies), C2R (Crisis to Resilience), and EES (Environmental and Economic Sustainability). Every one of Tearfund's projects contributes to and aligns with one (or more) of these priorities. Therefore, each evaluation report is assigned to one or more of the groups depending on the nature of the project that is being evaluated.

The primary goal of deductive coding is to find excerpts from evaluation reports that describe the differences (positive or negative impact) that Tearfund's work has made to people's lives and communities. The human coder starts with a codebook in mind to help guide the process, but will subsequently apply one or many codes from the codebook to categorize excerpts. Tearfund uses two codebooks: (a) their internally developed framework for understanding, working toward and measuring well-being, the Light Wheel[4]; and (2) the United Nations' Sustainable Development Goals[5] (SDGs). The excerpts are also coded for whether the difference that Tearfund has made is positive or negative and if the excerpts provide supporting evidence for impact. The deductive process poses a particularly challenging problem for applying LLMs as the task is not only to assign codes, but also extract suitable text excerpts. Additionally, evaluators will often make the same point or state the same conclusion more than once. The human coder, however, will typically only code each point/argument once.

In the case of inductive coding, Tearfund staff analyze each 'corporate priority group' of reports, identify patterns and themes that emerge naturally, and thereby uncover learning points that can inform future programs and the strategic direction of each of the corporate priorities.

### 3.2 Datasets

To assess the capabilities of using LLMs for performing qualitative analyses within the context of Tearfund's meta-synthesis process, we obtained example evaluation reports and outputs of the coding process. These were structured into three datasets (shown in Table 1) to assist with validating the proposed scenarios. The 'Codebook' column specifies the coding framework applied to each report; the 'Language' column indicates the language in which each report is written; 'Word Count' represents the total number of words in each report, excluding sections, such as introduction, methodology, and annexes; 'Deductive Coding Excerpts Count' shows the number of excerpts quoted through deductive coding methods; and 'Corporate Priority' specifies which corporate priority group each report was assigned to.

Dataset 1 consists of some of the evaluation reports that were included in the 2023 meta-synthesis, already analyzed and coded by Tearfund's expert coders. The focus of this meta-synthesis was on well-being,[6] so these evaluations had been coded with the Light Wheel codebook. Datasets 2 and

[2] Tearfund (site visited: 04/11/2024): https://www.tearfund.org/

[3] Bond Evidence Principles (site visited: 04/11/2024): https://www.bond.org.uk/resources/evidence-principles/

[4] Tearfund Light Wheel toolkit (site visited: 04/11/2024):https://learn.tearfund.org/en/resources/series/the-light-wheel-toolkit

[5] United Nations' Sustainable Development Goals (site visited: 04/11/2024): https://sdgs.un.org/goals

[6] Aiming to answer the following question: 'According to evaluators, how does Tearfund's work contribute to improving people's well-being?'.

**Table 1** Datasets used in these experiments

| | Report Name | Codebook | Language | Word Count | Excerpts Count | | Corporate Priority |
|---|---|---|---|---|---|---|---|
| | | | | | Deductive Coding | Inductive Coding | |
| Dataset 1 | | | | | | | |
| 1 | Savings groups and CCMP, Cote d'Ivoire | Light Wheel | English | 6691 | 27 | 23 | CCT, EES |
| 2 | RPS integrated programming, Burundi | Light Wheel | English | 5868 | 18 | 26 | RPS |
| 3 | Sangasangai (QuIP), Nepal | Light Wheel | English | 6550 | 30 | 13 | CCT |
| 4 | Emergency WASH, DRC | Light Wheel | English | 14,519 | 26 | 42 | C2R |
| 5 | Seed multiplication for small farmers, DRC | Light Wheel | French[a] | 1515 | 12 | 0 | EES |
| | | Total | | | 113 | 94 | |
| Dataset 2 | | | | | | | |
| 1 | CCT, Bangladesh | SDGs | English | 5471 | 10 | Not yet coded | CCT |
| 2 | Addressing sexual exploitation and abuse, Zimbabwe | SDGs | English | 7257 | 7 | Not yet coded | RPS |
| 3 | Climate and community transformation, India | SDGs | English | 5857 | 14 | Not yet coded | EES |
| 4 | Mental health and psychosocial support, Philippines | SDGs | English | 6005 | 6 | 8 | C2R |
| 5 | Triple Nexus project, Eurasia and North Africa | SDGs | English | 9384 | 12 | 8 | EES, RPS, C2R |
| | | Total | | | 49 | 16 | |
| Dataset 3 | | | | | | | |
| 1 | WASH Project, Eurasia and North Africa | SDGs | English | 7933 | Not yet coded | Not yet coded | |
| 2 | Community transformation through capacity building, India | SDGs | English | 7825 | Not yet coded | Not yet coded | |
| 3 | RNCDP, Bangladesh | SDGs | English | 4478 | Not yet coded | Not yet coded | |
| 4 | Tearfund Innovation Fund, Zimbabwe | SDGs | English | 9206 | Not yet coded | Not yet coded | |
| 5 | Trash to Cash, Nigeria | SDGs | English | 12,682 | Not yet coded | Not yet coded | |

[a]One report is written in French to test GPT's multilingual capabilities

3 consist of evaluation reports that are part of the ongoing 2024 meta-synthesis, some of which had already been coded at the time of our study. For the first time, as part of this meta-synthesis, the Tearfund team is focusing on the UN's SDGs,[7] meaning that the SDG codebook was applied to the evaluations in datasets 2 and 3.

As a pre-trained foundation model, the LLM already possesses knowledge of the SDGs. Therefore, we use datasets 1 and 2 to assess performance when coding with different codebooks.

## 3.3 Experimental setup

### 3.3.1 Human–LLM collaborative framework

In this study, we designed our experiments around two scenarios, breaking the human coding process into distinct tasks to facilitate prompt design (Noring et al. 2024). This allowed us to assess GPT-4o for deductive and inductive coding (see Fig. 1). In Scenario 1, the LLM acts as a *validation* tool, with its coding being compared to human expert coding to assess accuracy and consistency. This scenario emphasizes the LLM in a supporting role, where humans perform the primary coding tasks, and the LLM validates their work. Although the LLM offers validation feedback, humans retain full control over any final decisions.

In contrast, Scenario 2 positions the LLM as the *initial annotator*, taking on a more active role. This process starts with the human coder providing instructions and examples to establish a coding framework. The LLM performs the

[7] Aiming to answer the following question: 'According to evaluators, how is Tearfund's work contributing to the achievement of the SDGs?

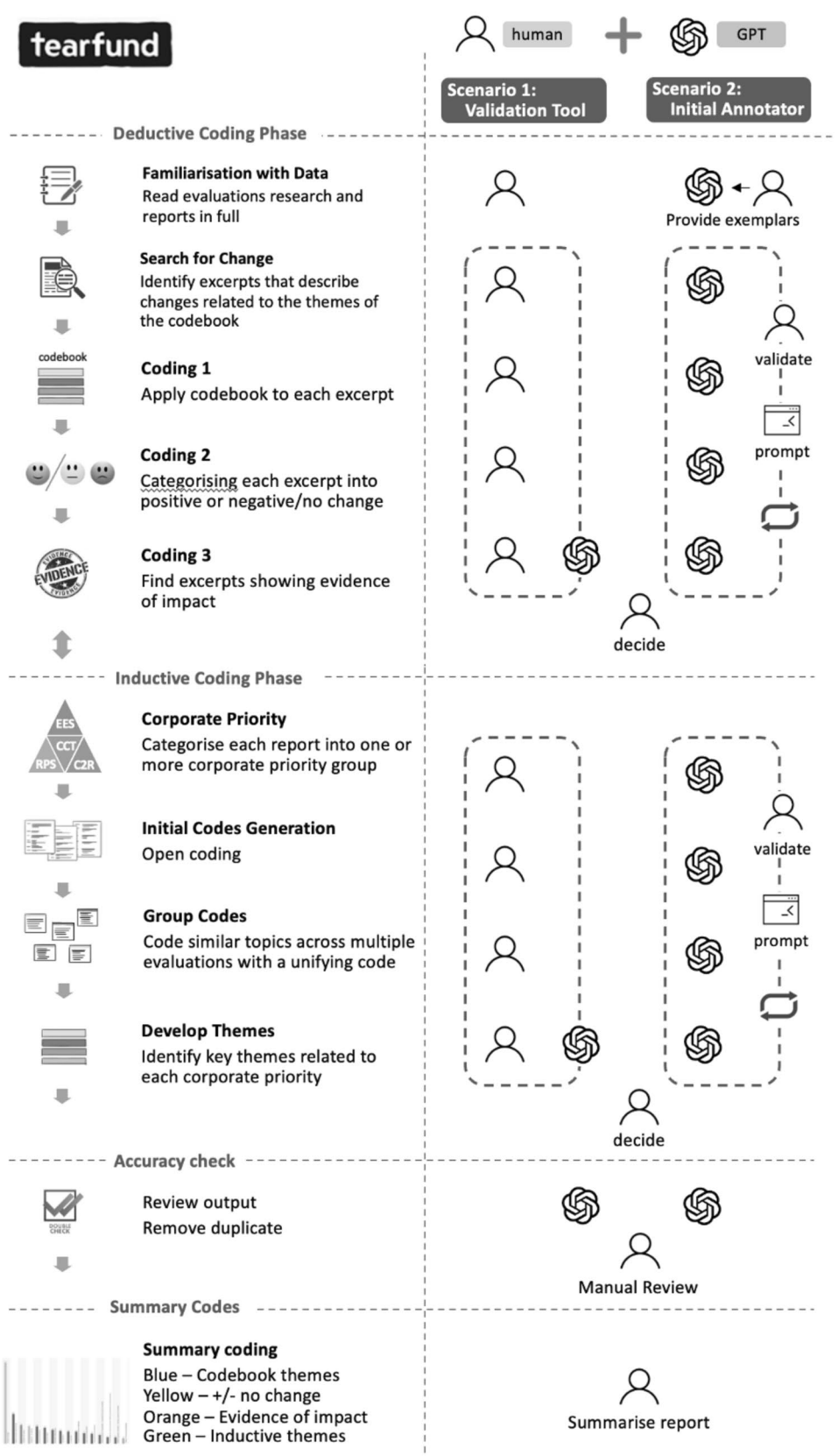


**Fig. 1** Human–LLM collaborative framework

initial coding, and its outputs are then iteratively validated and refined by a human judge. This scenario involves an iterative refinement process to train the LLM to emulate human coders. Together, these scenarios establish a human–LLM collaborative framework similar to (Tai et al. 2024; Turobov et al. 2024; Dai et al. 2023).

We validate these scenarios through an empirical study of LLMs using Tearfund documents and coding, along with inputs from Tearfund staff and their review of the outputs. In Scenario 1, the outputs were compared and assessed manually by the first author using multiple metrics to evaluate the performance of the LLM. For both Scenarios 1 and 2, the expert coder from Tearfund was used to evaluate outputs and provide feedback on the utility of LLMs in supporting the meta-synthesis workflow. This study received ethical approval from the University of Sheffield.

#### 3.3.2 GPT-4o and GPT-4o-mini

We use GPT-4o and GPT-4o-mini from OpenAI for deductive coding, while GPT-4o is exclusively used for inductive coding due to its higher requirements. These models represent the current state-of-the-art. OpenAI's ChatGPT platform is used for pilot runs because of its intuitive interface, which facilitates rapid testing and iterative refinement. For enhanced data security and adjusted temperature setting, the main experiments are conducted on Azure and OpenAI platforms. Given the differing token limitations and policies for request increases, GPT-4o is run in Azure AI Studio; while GPT-4o-mini is operated through OpenAI. Additionally, we benchmark GPT-4o's performance across both environments, finding that the results from each platform are consistent within an acceptable range of variability. Using GPT-4o for deductive coding validation on Code 1 (theme classification) as a benchmark, we observed the following: on the Light Wheel dataset, Cohen's kappa scores averaged 0.62 ($sd = 0.0224$) on Azure and 0.62 ($sd = 0.0273$) on OpenAI; on the UN SDGs dataset, Cohen's kappa scores averaged 0.65 ($sd = 0.0270$) on Azure and 0.67 ($sd = 0.0273$) on OpenAI. These results demonstrate stable and reproducible model performance across datasets and deployment environments.

For LLMs, temperature setting greatly influences the outcome of the model's responses with a higher value resulting in more random and creative outputs, while a lower value leads to more deterministic and reproducible results (Ekin 2023). In this study, we use a temperature setting of 0 for deductive coding to enhance consistency and ensure reproducibility (Chew et al. 2023; Dai et al. 2023; Hou et al. 2024; Xiao et al. 2023). For inductive coding, we aimed to balance creativity and precision by setting the temperature to 0.5, similar to De Paoli (2023).

### 3.4 Deductive coding

For deductive coding, we assess two main aspects: (1) *coding validation*, and (2) *excerpt extraction*. In the case of coding validation, we run the GPT model 10 times and report the average of results to take into account variability in the outputs. For this experiment, given the excerpts already identified by the human coder and assigned codes, we use GPT to validate them based on three levels of coding (see Table 2). This is necessary to help break the problem down into discrete tasks for guiding the model during prompting. In practice, however, the human coder undertakes all levels of coding during their analyses.

The first level (Code 1) evaluates how well GPT can assign 1 or many codes from the Light Wheel codebook or UN SDGs. The outputs are compared with the codes manually assigned using Cohen's kappa (Cohen 1960), a measure of inter-rater reliability (IRR) and previously used to evaluate LLMs for thematic analysis by Xiao et al. (2023). The score falls in the range from -1 to 1, where 0 represents the amount of agreement that can be expected from random chance, and 1 represents perfect agreement between raters. Since Code 1 is a multi-label classification, we use instance-based kappa to provide an overall measure of agreement by pooling the decisions across all instances and codes. The next level (Code 2) is similar to sentiment analysis and assesses whether excerpts express positive, negative or no change. Finally, at the third level (Code 3), we assess

**Table 2** Example levels of codes for deductive coding

| Theme or code (Code 1) | Evidence of change (Code 2) | Evidence of impact (Code 3) | Excerpt Number and First 20 Words |
|---|---|---|---|
| SDG Goal 1 End poverty, SDG Goal 8 Economic growth and decent work | Positive change | No | 3:10 The provision of Income Generating Activity (IGA) support has proven instrumental in enhancing participants' income levels. This support led… |
| SDG Goal 3 Healthy lives and well-being | Negative/ no change | Yes | 6:3. This data suggests that depression is not a key issue for community members in Bangued. Figure 2 also shows… |

whether the excerpt represents evidence of impact. Additionally, we record the time to execute the prompt (wall clock time) and hallucination rate (the number of runs not following instructions exactly / the total number of runs). The higher the hallucination rate, the more unstable the model.

To assess the ability of the GPT model to identify relevant excerpts (excerpt extraction), we provide the full text of an evaluation report within the prompt and compare the outputs with the excerpts previously identified (and coded) manually. The GPT outputs are evaluated using precision and recall (as also used by Dai et al. (2023)). Precision captures the proportion of relevant excerpts identified by GPT. Recall evaluates how many of the excerpts identified by Tearfund's experts were also captured by GPT. We applied leniency in the matching to accommodate cases where the exact text did not match, but the overall meaning of the excerpt was the same. For example, the human coder extracted the excerpt "Because of CCMP I have built 2 stores. I increased the capital in my DIY store, and bought land" and GPT extracted the excerpt "[I liked the] mobilization of resources. Because of CCMP I have built 2 stores. I increased the capital in my DIY store, and bought land." These were viewed as a match.

### 3.5 Inductive coding

Given that inductive coding is data-driven and allows themes to emerge naturally, even in the validation scenario, the full text was inputted to GPT-4o via the prompt. Cosine similarity was used to measure the alignment between GPT's coding and the human coding. The analysis was conducted on datasets 1 and 2, comprising 4 reports and 2 reports respectively. To compare the themes identified using GPT-4o against the human outputs, we used OpenAI's embedding model (text-embedding-ada-002) to create a semantic representation of the themes and codes prior to calculating similarity. This approach is based on the method described by Dai et al. (2023). Additionally, categorization against Tearfund's four corporate priorities was assessed by Agreement Rate, calculated using a weighted scoring system that accounts for exact matches and varying degrees of partial alignment (see also Sect. 4.2).

### 3.6 Human feedback

Tearfund staff were involved in the experimental design and reviewing and refinement of the LLM's outputs. For both Scenarios 1 and 2, we gathered quantitative and qualitative feedback on their overall assessment of the LLMs (as previously adopted by Meng et al. (2024)). Similar to Yan et al's (2024) study, four evaluation questions were developed, with the first three using a 1–5 rating scale (where 1 indicates "low" and 5 indicates "high") to measure how accurately the models identified themes and assigned codes (accuracy), how efficient the models would be in assisting with the coding process (efficiency), and the extent to which the human coder would trust the outputs (trust). The fourth open-ended question was used to gather additional feedback and suggestions for using LLMs in the coding process. A detailed description of human-in-the-loop evaluation process is provided in Appendix A, and the detailed Likert scale that was used to evaluate accuracy, efficiency, and trust is provided in Appendix B.

### 3.7 Prompt strategy

To use LLMs for thematic analysis, we employ natural language prompts as input. Tailoring prompts effectively is essential to maximize the utility of LLM and as such, our prompt strategy is grounded in three core principles: decomposing tasks, crafting specific prompts, and validating outputs (Noring et al. 2024). This process involves a basic prompt–response loop, whereby a prompt is provided to the AI model and the model generates a response referred to as a 'Completion.' For complex tasks within Tearfund's context, we designed step-by-step prompt templates[8] to guide the LLM through a structured workflow. Each template begins with a meta-prompt that defines the LLM's role and overarching objective, followed by task-specific instructions and requests for rationale for inductive coding. These templates are iteratively refined to ensure accuracy and relevance and structured as follows:

- **Deductive coding:** Prompt 1 (Define the task and role) → Prompt 2 (Provide the codebook) → Prompt 3 (Instruct deductive coding) → Prompt 4 (Input data for coding) → Prompt 5 (Validate the results)
- **Inductive coding:** Prompt 1 (Define the task and role) → Prompt 2 (Conduct open coding) → Prompt 3 (Search for themes) → Prompt 4 (Input data for coding) → Prompt 5 (Validate the results)

Figure 2 illustrates the experimental workflow for deductive coding where we bridge strategy and practice by documenting the prompt patterns and engineering techniques used in this case study. It highlights the interaction between human oversight and LLMs through examples of human 'prompt' and LLM 'completion' loops, along with key learnings (both positive and negative) ascertained during experiments. The workflow comprises five distinct stages, incorporating specific prompt patterns and concluding with manual evaluation to ensure quality and alignment with objectives.

[8] For detailed prompt designs used in the experiments, see (site visited: 21/11/2024): https://github.com/pauldclough/tearfund-metasynthesis

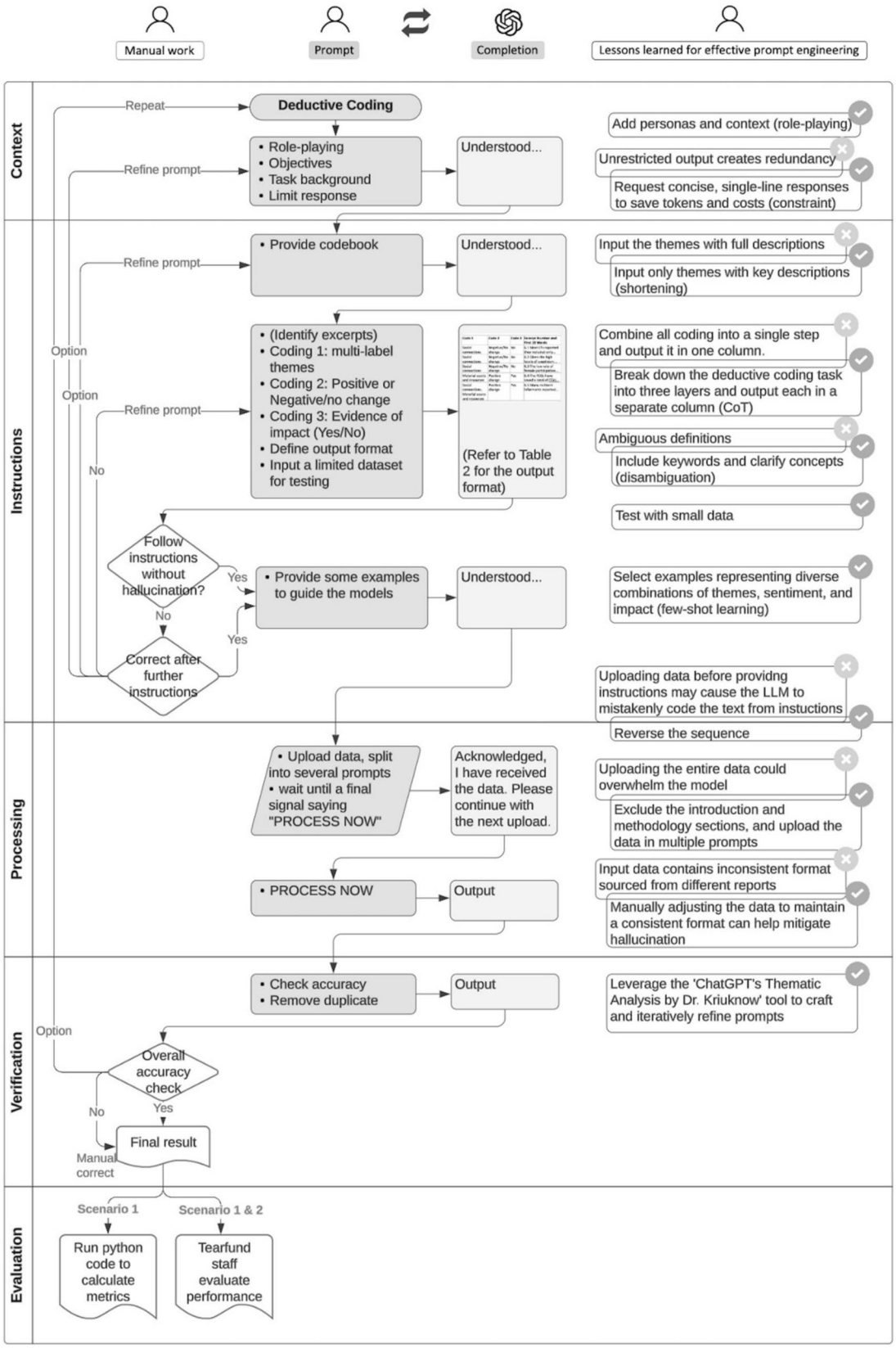


**Fig. 2** Deductive coding workflow with insights for effective prompt engineering

**Table 3** Overview of results for deductive coding (coding validation)

| Codebook | Dataset 1 (Tearfund Light Wheel) | | | | Dataset 2 (UN SDGs) | | | |
|---|---|---|---|---|---|---|---|---|
| Model | GPT-4o Zero-Shot | GPT-4o Few-Shot | GPT-4o-mini Zero-Shot | GPT-4o-mini Few-Shot | GPT-4o Zero-Shot | GPT-4o Few-Shot | GPT-4o-mini Zero-Shot | GPT-4o-mini Few-Shot |
| Prompt Execution (Mins:Secs) | 01:55 | 01:58 | 02:02 | 02:27 | 01:00 | 01:16 | 01:08 | 01:30 |
| Hallucination Rate | 23.1% | 16.7% | 66.7% | 50.0% | 9.1% | 9.1% | 25.0% | 9.1% |
| Cohen's kappa | | | | | | | | |
| Themes (Code 1) | 0.62 | 0.61 | 0.46 | 0.45 | 0.64 | 0.65 | 0.58 | 0.53 |
| Sentiment (Code 2) | 0.91 | 0.92 | 0.91 | 0.93 | 0.95 | 0.95 | 0.91 | 0.92 |
| Evidence of Impact (Code 3) | -0.02 | -0.02 | 0.01 | -0.01 | -0.08 | -0.08 | -0.09 | -0.09 |

While the diagram visually represents the deductive coding workflow, inductive coding follows a similar structure, with notable differences in the 'Instructions' stage. The five stages are:

- **Context**: Uses the Persona-Instruction-Context (PIC) pattern to define the AI's role, overarching tasks, and provide relevant background information (Noring et al. 2024).
- **Instructions**: Applies the Task-Action-Guideline (TAG) pattern to break the coding task into smaller, actionable steps (Noring et al. 2024). For deductive coding, this involves applying a codebook and following specific coding instructions. In contrast, inductive coding emphasizes open coding to generate initial codes, followed by grouping similar codes to identify emerging themes.
- **Processing**: Executes instructions by inputting text for coding and generating outputs based on prior steps.
- **Verification**: Uses the Fact-Check-List (FCL) pattern (White et al. 2023), incorporating follow-up prompts for self-checks and a human validation process. Tools, such as 'ChatGPT's Thematic Analysis by Dr. Kriuknow'[9] can be leveraged to iteratively refine and enhance prompts, ensuring they effectively achieve the desired outputs.
- **Evaluation**: Conducts manual review and computes multiple metrics using Python code for Scenario 1 and gathers 1–5 scale ratings for accuracy, efficiency, and trust, along with qualitative assessments by Tearfund staff for both scenarios.

[9] ChatGPT's Thematic Analysis by Dr. Kriuknow (site visited: 30/11/2024):https://chatgpt.com/g/g-hAcekuQIB-thematic-analysis-by-dr-kriukow

# 4 Results

## 4.1 Deductive coding

**Coding validation.** The results for the coding validation aspect of the deductive coding process are shown in Table 3. This shows the results obtained using GPT-4o and GPT-4o-mini with zero and few-shot learning across the two codebooks.

When classifying themes (Code 1), the GPT-4o models obtain the highest kappa score of 0.65 with few-shot learning and the UN SDGs codebook. Similar scores are also obtained with the Light Wheel codebook. Given that Cohen's kappa between 0.61 and 0.80 indicates a substantial agreement (McHugh 2012), this suggests that GPT-4o demonstrates a substantial agreement with human experts for Code 1 even though this involves a multi-label classification. In contrast, GPT-4o-mini shows only a moderate agreement with the gold standard, indicating that smaller language models may struggle with more complex classification tasks. We also observed more variation and instability across runs when using GPT-4o-mini. For example, when using the UN SDG codebook and the few-shot learning, the model achieved an average kappa score of 0.53 ($sd = 0.10$), but ranged from 0.39 to 0.63. In contrast, the few-shot GPT-4o model average kappa score was 0.65 ($sd = 0.03$) ranging from 0.58 to 0.69.

While both models are prone to hallucinations, GPT-4o-mini is more likely to drift off-task, requiring frequent user correction. This is particularly evident when performing deductive coding on a sequence of excerpts where GPT may omit certain excerpts, fabricate non-existent ones, or present them in the wrong order, disrupting the intended sequence. For instance, when tasked with coding excerpts 16.10, 16.11, 16.13, 16.14, and 16.15, GPT-4o-mini might omit 16.11 and 16.13, fabricate a non-existent 16.12, or output the sequence out of order during different runs. We also find that the LLM may misinterpret thematic codes, replacing the original theme, such as "Care of the environment"

with a similar, but incorrect (or invented) one, "Sustainable environment." These inconsistencies highlight the need for more oversight when using GPT-4o-mini compared to the more reliable GPT-4o.

Results show that in the case of classifying sentiment of the excerpts (Code 2), all models perform with high levels of agreement—GPT-4o giving the highest kappa score of 0.95. This suggests that the task is far easier for the LLMs to perform. Identifying evidence of impact (Code 3) resulted in the lowest agreement with human coding with the highest kappa score being 0.01 (GPT-4o-mini zero-shot). The LLMs likely struggled with this task due to the complexity and variability in how impact is evidenced across Tearfund's reports, requiring human intervention for accurate interpretation.

We observe no significant difference in performance between dataset 1 (Light Wheel) and dataset 2 (SDGs) when classifying themes (Code 1) using GPT-4o. However, GPT-4o-mini's reduced performance on dataset 1 indicates added difficulty for smaller models. For dataset 1, the models were provided with a detailed codebook and clear definitions centered around the concept of well-being, developed by Tearfund. In contrast, for dataset 2, the models were instructed to code based on the UN's 17 SDGs, which are part of the general pre-trained data. Despite these different approaches, GPT-4o performed comparably across both datasets, while GPT-4o-mini struggled on dataset 1. Additionally, we also observed that the hallucination rate is far higher for dataset 1, particularly in the case of zero-shot and GPT-4o-mini. This difference in hallucination rates is likely because Tearfund's Light Wheel framework, being an internally developed tool, is less represented in the public training data of LLMs. In contrast, the UN SDGs are globally established and commonly included in model training corpora, making them easier for LLMs to handle. As a result, the models exhibited greater familiarity and lower hallucination when coding against the SDGs compared to the Light Wheel. This may suggest that the use of coding examples and a more capable model is required when using a custom codebook.

**Excerpt Extraction.** The results of using GPT-4o for extracting relevant excerpts from the evaluation reports are shown in Table 4. This shows individual results for precision and recall across the 10 reports from datasets 1 and 2 that contain 162 excerpts identified by the human coder. The average precision score of 0.41 suggests that around 41%

**Table 4** Overview of results for deductive coding (excerpt extraction)

| File | Avg | LW_1 | LW_2 | LW_3 | LW_4 | LW_5 | SDGs_1 | SDGs_2 | SDGs_3 | SDGs_4 | SDGs_5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Precision | 0.41 | 0.58 | 0.57 | 0.56 | 0.35 | 0.48 | 0.31 | 0.25 | 0.50 | 0.24 | 0.31 |
| Recall | 0.53 | 0.70 | 0.44 | 0.63 | 0.31 | 0.83 | 0.50 | 0.29 | 0.57 | 0.67 | 0.33 |

**Table 5** Cosine similarity for themes and CP categorization for inductive coding

| | Report Name | Corporate Priority | | Comparison | |
|---|---|---|---|---|---|
| Dataset 1 | | | | | |
| | | Tearfund | GPT | Match Type | Cosine Similarity |
| 1 | Savings groups and CCMP, Cote d'Ivoire | CCT, EES | CCT, EES | Exact match | 0.7537 |
| 2 | RPS integrated programming, Burundi | RPS | RPS, CCT | Partial match[1] | 0.7819 |
| 3 | Sangasangai (QuIP), Nepal | CCT | CCT, RPS, C2R | Partial match[1] | 0.7483 |
| 4 | Emergency WASH, DRC | C2R | C2R, EES | Partial match[1] | 0.7678 |
| 5 | Seed multiplication for small farmers, DRC | EES | EES, C2R | Partial match[1] | n/a |
| | Average | | | | 0.7629 |
| Dataset 2 | | | | | |
| | | Tearfund | GPT | Match Type | Cosine Similarity |
| 1 | CCT, Bangladesh | CCT | CCT, EES | Partial match[1] | n/a |
| 2 | Addressing sexual exploitation and abuse, Zimbabwe | RPS | RPS, CCT, C2R | Partial match[1] | n/a |
| 3 | Climate and community transformation, India | EES | CCT, C2R, EES | Partial match[2] | n/a |
| 4 | Mental health and psychosocial support, Philippines | C2R | C2R, CCT, RPS | Partial match[1] | 0.7955 |
| 5 | Triple Nexus project, Eurasia and North Africa | C2R, RPS, EES | C2R, RPS, EES | Exact match | 0.7779 |
| | Average | | | | 0.7867 |

Partial match[1]: Partial match – primary CP aligned

Partial match[2]: Partial match – primary CP not aligned

of excerpts identified by the LLM as relevant actually were (true positive). On the other hand, the average recall of 0.53 suggests that just over half of all excerpts that the human coder had identified were also found by the LLM.

The results align with the feedback from Tearfund staff for dataset 3 (uncoded, SDGs) where they noted that GPT-4o tended to flag too many irrelevant excerpts. For instance, this applied to about 8 out of 23 coded excerpts from the India evaluation and 9 out of 19 from the Bangladesh evaluation.

### 4.2 Inductive coding

The results from the inductive coding experiments are shown in Table 5. In the case of assigning corporate priorities (CPs), when compared with the categories assigned by the human coder, GPT-4o is able to align correctly with all or some of them across the 10 evaluation reports. If we weigh an exact match as 1, a partial match (primary CP aligned) as 0.75 and partial match (primary CP not aligned) as 0.5, then the agreement rate is 80% for dataset 1 and 75% for dataset 2. As shown in Table 5, GPT-4o also tends to assign more corporate priority categories per report, leading to a higher partial match rate. However, despite assigning multiple categories, GPT-4o accurately identifies the primary CP in most cases.

In the case of measuring thematic alignment between GPT-4o's generated codes/themes and those created by the human coder, we obtain a cosine similarity of 0.7629 for dataset 1 and 0.7867 for dataset 2. For example, human coding identified "Social norms interventions: Joint engagement of men and women," while GPT-4o organized-related codes, such as "Challenging gender norms," "Positive masculinity," and "Breaking the silence", under the theme of "Gender Norms and Equality." This suggests that GPT-4o is proficient at identifying and coding the key semantic elements in the evaluation reports, closely aligning with human coders. However, we also observed cases of hallucination where GPT-4o altered the meaning of acronyms. For example, changing "CCT: Church and Community Transformation" to "Climate Change and Transformation," and "RPS: Reconciled, Peace-filled Societies" to "Resilience and Protection Services."

### 4.3 Overall evaluation

The qualitative feedback provided by Tearfund staff provides insights into GPT-4o's performance across the dimensions of accuracy, efficiency, and trust (see Fig. 3).

Overall, GPT-4o performed better in accuracy and efficiency during inductive coding compared to deductive coding, but scores lower for trust: *"I have given a low trust score because I do not trust GPT to provide useful themes on its own, but the efficiency score is somewhat higher because it could add value as a brainstorming tool without it taking much extra time."*

For deductive coding, the expert gave lower scores because of GPT's tendency to over-generate excerpts: *"While the GPT models have identified and coded excerpts that meet these criteria (and my sense is that they have not missed or overlooked many), they have also coded a large number of excerpts that do not meet the criteria. This lowered my accuracy score. Unless I or another (human!) analyst were to review all excerpts and remove those that are not relevant, I think this has the potential to produce results that are misleading for us at Tearfund, thus lowering the scores for efficiency and trust."*

Overall, based on the feedback in the open question, the following strengths of GPT were identified: accurate assignment of SDG codes, accurate sentiment analysis, efficient inductive coding process, and useful as a brainstorming tool in inductive coding. Additionally, GPT offers potential for substantial time reduction in deductive coding. However, this time-saving benefit is significantly diminished by the need for extensive human review to ensure accuracy and reliability as noted in the feedback: *"if we wanted to maintain our level of confidence in the results, I feel it would require*

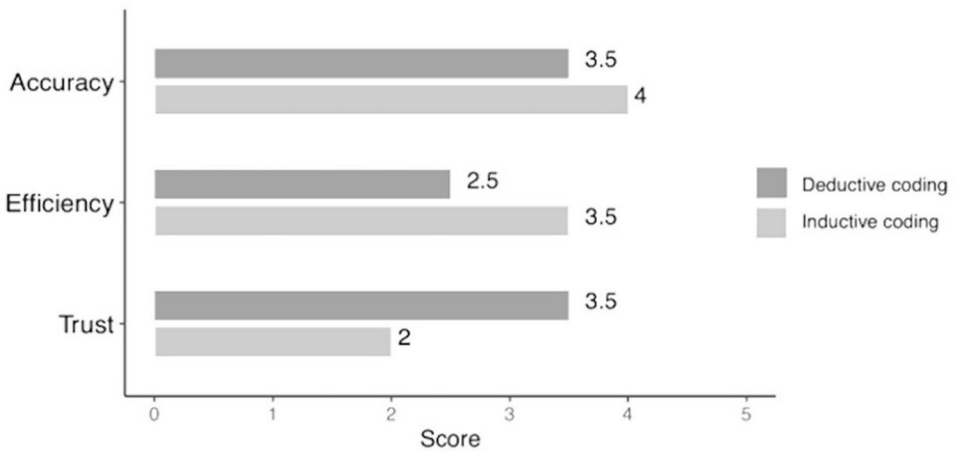


**Fig. 3** Comparison GPT-4o for deductive and inductive coding

**Table 6** Summary of the roles of GPT-4o in deductive and inductive coding

| | Scenario 1 (as a validation tool) | Scenario 2 (as an initial annotator) |
|---|---|---|
| Deductive coding | (+) Provide a more objective perspective | (-) Too many irrelevant excerpts may mislead the interpreter |
| Inductive coding | (-) Low trust in GPT provided themes | (+) High accuracy and useful as a brainstorming tool |

*me to do a lot of reviewing of GPT's work, giving feedback and asking it to improve, which would significantly reduce the amount of time I could save."* Other weaknesses were also identified: low accuracy in identifying relevant excerpts, misleading results without human review, difficulty in identifying evidence of impact, broad themes in inductive coding are not always useful, and low trust in GPT to provide useful themes on its own.

## 5 Discussion

We discuss our findings within the context of the initial research questions and our overall suggestions for using LLMs to support thematic analysis.

### 5.1 Using LLMs for thematic analysis

(RQ1): How well can LLMs perform deductive and inductive coding tasks?

We evaluated the effectiveness of GPT-4o for thematic analysis within two scenarios. In the case of validating existing deductive analysis, results suggest that GPT-4o could achieve a substantial agreement with human coders for theme identification (Code 1) and sentiment classification (Code 2). This aligns with previous findings where the consistent improvements of LLMs have been translated into improved thematic analysis results. For example, Xiao et al. (2023) used GPT-3, Gao et al. (2024) and Dai et al. (2023) applied GTP- 3.5, and Zhang et al. (2024) and Sinha et al. (2024) employed GPT-4 for thematic analysis, with each iterative model delivering improved performance.

However, identifying evidence of impact (Code 3) proves to be more challenging and discussions with Tearfund staff suggest that this difficulty may stem from the varying methodologies used by different evaluators in the reports thereby making it difficult for GPT-4o to consistently identify evidence of impact without a uniform standard to guide its analysis. These results are consistent with the study by Suter and Meckel (2024), which found that GPT-4 achieved a substantial agreement with human codes for classifying news articles; however, performance declined when dealing with more complex constructs that require subjective interpretation.

For extracting excerpts, GPT-4o achieved a precision of 41% and a recall of 53% compared to a human ground truth, suggesting a tendency to identify too many irrelevant excerpts, which may mislead researchers. Tearfund's feedback underscores the need to balance GPT's advantages with caution to avoid misleading results. This highlights the critical role of expertise. Experienced coders can easily filter out irrelevant excerpts, but for novice coders, such distractions may shift their focus and lead to misinterpretation. It also indicates that GPT cannot yet fully automate the coding process and still necessitates human oversight to ensure accuracy and relevance as shown by Zhang et al. (2024).

Tearfund also recognized that for the SDGs dataset, GPT's labeling provides valuable insights that aid in reassessing the codes, reflecting its extensive pre-trained knowledge in this area. This finding aligns with previous studies, which suggest that using LLMs as validation tools can help uncover new insights (Meng et al. 2024; Zhang et al. 2024). In contrast, for the Light Wheel dataset, where domain-specific expertise plays a greater role, GPT's contributions were less impactful, highlighting the potential limitations of LLMs in domains requiring niche expertise.

For inductive coding, GPT-4o achieved moderate agreement rates in categorizing corporate priorities although it tended to over-assign categories. The model also demonstrated strong semantic alignment with human-generated codes and the ability to explain its rationale for categorizing and coding. This capability fosters transparency and interpretability, supporting Arlinghaus et al. (2024)'s findings that a clear rationale and transparency strengthen credibility and reproducibility. Tearfund further noted GPT-4o's higher accuracy and efficiency in inductive coding compared to deductive tasks. This aligns with findings from Bijker et al. (2024), which suggest that inductive coding generally outperforms deductive coding. Despite these strengths, qualitative feedback from Tearfund staff raised concerns about GPT-4o's tendency to produce overly broad themes and its lack of contextual understanding, ultimately diminishing their trust in directly adopting the model's suggested themes or codes. Similarly, in the health sector, Mannstadt et al. (2024) reported that GPT-4 efficiently generates comparable themes and survey questions, but lacks the precision of human-generated outputs. Nevertheless, GPT-4o proves useful as a brainstorming tool, offering a good starting point for human coders in inductive coding tasks. Table 6 summarizes

the use of GPT in deductive and inductive coding within Tearfund's context.

### 5.2 Collaborative human–LLM framework

(RQ2): How can LLMs be implemented within an existing thematic analysis workflow?

In this study, we proposed a collaborative human–LLM framework for augmenting the existing coding workflows. In line with prior research (De Paoli 2023; Gao et al. 2024; Lee et al. 2024; Zhang et al. 2024), we agree that LLMs complement human skills rather than replace them. In our case study for deductive coding, GPT-4o can surface insights that humans might miss and expedite the coding process, offering a layer of verification to improve coding accuracy and consistency. However, humans retain control over the accuracy of the results and maintain the final decision-making authority. Similarly, for inductive coding, GPT-4o demonstrates efficiency by rapidly sorting and categorizing large datasets, generating useful code suggestions in under 4 min per report. Despite this, expert interpretation is still necessary to analyze the results and make nuanced judgments (Khan et al. 2024; Goyanes et al. 2025). In both cases, human oversight is indispensable for ensuring the quality and reliability of the results.

Moreover, we emphasize the importance of expertise, or domain knowledge, for effectively leveraging LLMs in thematic analysis for three key reasons. First, LLMs achieve their full potential with well-crafted prompts, which require domain knowledge to provide appropriate background, detailed instructions, and iterative refinement. Second, users play a pivotal role in reviewing and validating GPT's outputs. While GPT can offer rational suggestions, it lacks the contextual understanding necessary for fully informed decisions, making user judgment indispensable. Third, LLMs are prone to hallucinations, often presenting false information with confidence. It falls on users to discern which insights to trust. Striking a balance between leveraging GPT's ability to uncover overlooked insights and managing the risk of misleading conclusions is key to effective collaboration between humans and LLMs.

### 5.3 Key takeaways

**Prompt engineering in thematic analysis.** Prompt engineering plays a crucial role in the successful use of LLMs for thematic analysis (De Paoli 2023; Hou et al. 2024; Turobov et al. 2024; Zhang et al. 2024). Techniques, such as role-playing, chain-of-thought prompting, and in-context learning, enable LLMs to handle complex tasks more effectively. These methods help steer the model to generate relevant insights by providing structured guidance and helping it navigate through multifaceted problems.

**Incorporating domain expertise.** A key also to this work was utilizing knowledge from Tearfund's experts: from developing appropriate task workflows, through to designing prompts and evaluating LLM outputs. In particular, prompt design is highly dynamic, and designers need to adjust and iterate on their prompts based on the outputs generated by the model. Our findings support Zhang et al. (2025)'s idea that when domain expertise is combined with thoughtful prompt engineering, LLMs can support the thematic analysis process, although one needs to bear in mind their weaknesses.

**LLMs as complementary tools with human oversight.** GPT-4o can enhance the efficiency, accuracy, and scalability of qualitative coding. It acts as a validation tool in deductive coding, uncovering insights and speeding up the process, and as a brainstorming aid in inductive coding, generating quick code suggestions. However, human oversight is crucial for maintaining accuracy, contextual understanding, and correcting over-generalizations or mitigating biases in the output. This echoes Nguyen-Trung (2025) and Zhang et al. (2025)'s call for a balanced approach, cautioning against over-reliance on AI. Researchers must weigh the benefits of faster analysis against potential compromises in depth, insight, and trust.

**Choice of model**. Although the GPT-4o models are among the most capable to date, the smaller model struggled with the complexity of the task at hand and suffered from higher hallucination despite being cheaper. We also found the smaller model to be far less stable with more varied outputs each time the model was run compared to GPT-4o and benefit more from few-shot learning to reduce hallucinations.

**Efficiency vs. Accuracy trade-offs.** While GPT-4o may significantly speed up coding tasks, the time saved is offset by the need for human validation and correction. Human review is necessary to maintain accuracy and prevent the model from drifting off-task, which can extend the overall process time. There is also a need to learn effective prompting techniques that may also require substantial time commitment. All in all, AI is no 'silver bullet' solution.

### 5.4 Study limitations

This study has several limitations. First, few-shot learning results in negligible improvement or worse performance in theme classification compared to zero-shot learning. Despite selecting diverse examples, factors, such as overfitting, task complexity, unintentional bias, and differences in how LLMs process different codebooks, may affect the learning process. Further experimentation is needed to explore this issue in more depth. Second, prompt design remains a key challenge. While iterative improvements were made, further refinement is essential, particularly by incorporating domain expertise to

allow real-time adjustments. Lastly, the reliance on a single case study, Tearfund's evaluation meta-synthesis, limits the generalizability of the findings. Future research should test the framework across diverse sectors and contexts to assess its versatility and effectiveness. Additionally, the potential for using GPT models as evaluators, as suggested by Chiang & Lee (2023), offers a promising avenue for expanding the role of LLMs in qualitative research. This could also include their use in summarizing reports further enhancing the efficiency and depth of analysis in various fields.

## 6 Conclusions

In this work, we have explored the role of LLMs within a specific qualitative analysis workflow at Tearfund, a large UK-based charity. We propose two scenarios in which LLMs can support the manual coding process and test these through a series of experiments using representative documents and coding outputs, as well as Tearfund staff leading the analysis process. Results show that GPT-4o models can effectively serve as both an initial coding tool and a validation mechanism for inductive and deductive coding processes within an LLM-human collaborative framework. AI technologies, such as GPT-4o, are reshaping traditional workflows and as such introduces a dual learning curve: users must evolve their understanding of LLM capabilities, particularly mastering skills like prompt engineering; while LLMs themselves continue to improve and expand their capabilities through iterative updates. The synergy between human expertise and machine learning promises to further enhance productivity and effectiveness in thematic analysis and other knowledge-intensive tasks.

Further work will continue to investigate integrating LLMs into the coding workflows with the aim of streamlining the process. This may include further development of prompts and strategies for automatically validating the outputs, the use of new AI techniques, such as multi-agent AI and multimodal models, and upskilling the workforce. As the technologies improve, there will undoubtedly be many benefits of AI technologies within the charity sector beyond thematic analysis as long as these are balanced with the potential risks and challenges AI incurs.

## Appendix A. Human-in-the-Loop Evaluation for LLM Performance

The annotations used to evaluate LLM outputs were generated by a single expert coder at Tearfund with extensive experience in thematic analysis and internal evaluation standards. Given the domain-specific nature of the task, these expert-generated annotations served as the gold standard for performance assessment.

The LLM evaluation followed a human-in-the-loop methodology involving iterative collaboration between the first author and Tearfund staff:

- **Prompt Development and Testing:** The first author initially designed prompt templates and tested LLMs to generate outputs for both deductive and inductive coding tasks.
- **Expert Review and Feedback:** Tearfund's expert coder reviewed the LLM outputs to assess their thematic validity, contextual relevance, and consistency with Tearfund's internal evaluation standards. These reviews functioned as sanity checks, focusing on output plausibility, nuance, and alignment with coding expectations.
- **Iterative Refinement:** Based on expert feedback, the prompts were refined to improve clarity, specificity, and reduce hallucination. This included revising phrasing, adding context cues, and decomposing tasks to better match LLM capabilities.
- **Prompt Engineering as Craft:** Prompt design proved to be more an art than an exact science. The quality of LLM outputs was highly sensitive to prompt phrasing and task framing. To address this, we applied principles, such as chain-of-thought scaffolding, task decomposition, and domain-specific constraints, to iteratively optimize prompt performance.
- **Finalization:** This iterative process continued until the prompts consistently produced outputs that met the quality standards set by the expert coder. These final prompts were then used to generate outputs for formal evaluation against the expert-generated gold standard annotations.

## Appendix B. Qualitative Evaluation Criteria

A 5-point Likert scale (1 = Very Low, 2 = Low, 3 = Moderate, 4 = High, 5 = Very High) was applied to the dimensions of Accuracy, Efficiency, and Trust.

| Criterion | Description | 1 (Very Low) | 3 (Moderate) | 5 (Very High) |
|---|---|---|---|---|
| **Accuracy** | | | | |
| A1 Excerpt/Theme Relevance | Whether the LLM selected excerpts or generates themes directly related to the research question | Completely unrelated excerpts/themes | Half-relevant excerpts; basic understanding of topic | Perfectly aligned excerpts; precise theme capture |

| Criterion | Description | 1 (Very Low) | 3 (Moderate) | 5 (Very High) |
|---|---|---|---|---|
| A2 Theme Classification Accuracy | Whether themes, sentiments, and evidence of impact labels were correctly applied to excerpts | Consistent misclassifications and flawed labeling | Occasional errors with general correctness | Classifications match expert human coding precisely |
| A3 Theme Cohesion and Granularity | Whether the generated themes and subthemes were appropriately grouped and detailed, similar to human coding standards | Chaotic organization; meaningless analysis | Basic organization; acceptable but unsophisticated | Exceptional organization; optimal detail balance |
| A4 Error Impact | Whether errors in the output could seriously mislead users or affect conclusions | Critical errors leading to dangerous misinterpretation | Minor errors not affecting main conclusions | Virtually error-free analysis |
| **Efficiency** | | | | |
| E1 Setup Overhead | Time and effort required to create prompts and prepare model inputs compared to manual coding | Complex setup; negates time savings | Manageable setup; requires some technical knowledge | Near-instant setup with minimal effort |
| E2 Scalability | How easily and quickly the model processed multiple documents | System failures with multiple documents | Adequate handling with some limitations | Exceptional scaling with consistent speed |
| E3 Cleanup effort | Analyst time needed to review, edit and finalize model outputs | Complete reworking required | Moderate targeted fixes needed | Immediately usable output |
| E4 Net Time Saving | Overall reduction in analyst work hours after using the model | No time savings or increased workload | Moderate reduction | Transformative efficiency |
| **Trust** | | | | |
| T1 Confidence in Findings | Comfort level in using the model's results for internal communication or reporting | Unusable results | Suitable for internal discussion | Fully trusted |
| T2 Decision-Readiness | Suitability of the model's outputs for informing operational or strategic decisions | Unsuitable for any decisions | Acceptable for preliminary decisions with oversight | Excellent foundation for strategic decisions |
| T3 Transparency and Explanation Clarity | How clearly the model's reasoning and any limitations could be understood by human coders | Completely opaque reasoning | Basic transparency requiring effort to follow | Exceptional transparency; clear reasoning |
| T4 Cross-Document Consistency | Whether the model's outputs were consistent when applied to different evaluation reports | Wildly inconsistent across documents | Generally consistent with noticeable variations | Perfect consistency across all documents |

**Author contribution** All authors contributed to defining the research topic, designing the experiments, providing input on the analysis and interpretation of results, and reviewing and refining the manuscript. C.W. and P.C. developed the methodological framework and wrote the main manuscript text. C.W. conducted the experiments and prepared Figs. 1–3 under the guidance and supervision of P.C. and R.P. R.P. also provided Tearfund's coding workflows and datasets, and supported C.W. in refining prompts and validating the experimental outputs. R.M. initiated this case study project and ensured ethical data management.

**Data availability** No datasets were generated or analyzed during the current study.

## Declarations

**Conflict of interest** The authors declare no competing interests.



## References

Arlinghaus, C. S., Wulff, C., & Maier, G. W. (2024). Inductive Coding with ChatGPT-An Evaluation of Different GPT Models Clustering Qualitative Data into Categories. OSF Preprints. https://doi.org/10.31219/osf.io/gpnye.

Arvidsson, S., & Axell, J. (2023). Prompt engineering guidelines for LLMs in Requirements Engineering.

Bijker R, Merkouris SS, Dowling NA, Rodda SN (2024) ChatGPT for automated qualitative research: content analysis. J Med Internet Res 26:e59050

Braun V, Clarke V (2006) Using thematic analysis in psychology. Qual Res Psychol 3(2):77–101

Brown T, Mann B, Ryder N et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901

Chew, R., Bollenbacher, J., Wenger, M., Speer, J., & Kim, A. (2023). LLM-assisted content analysis: Using large language models to support deductive coding. *arXiv preprint* arXiv:2306.14924.

Chiang, C. H., & Lee, H. Y. (2023). Can large language models be an alternative to human evaluations?. *arXiv preprint* arXiv:2305.01937

Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Measur 20(1):37–46

Cooke, H. (2024). Mitigating LLM hallucinations in text summarisation. *LinkedIn*. https://www.linkedin.com/pulse/mitigating-llm-hallucinations-text-summarisation-henry-cooke-kymae/?trackingId=Q7ovVWlcQlSIaR6EXKR8XQ%3D%3D Accessed July 3, 2024

Crowe S, Cresswell K, Robertson A et al (2011) The case study approach. BMC Med Res Methodol 11:100. https://doi.org/10.1186/1471-2288-11-100

Dai, S.-C., Xiong, A., & Ku, L.-W. (2023). LLM-in-the-loop: Leveraging Large Language Model for Thematic Analysis. *arXiv preprint* arXiv:2310.15100.

De Paoli S (2023) Performing an inductive thematic analysis of semi-structured interviews with a large language model: an exploration and provocation on the limits of the approach. Soc Sci Comput Rev 42(4):997–1019

Do, H. J., Ostrand, R., Weisz, J. D., Dugan, C., Sattigeri, P., Wei, D., Murugesan, K., & Geyer, W. (2024). Facilitating Human-LLM Collaboration through Factuality Scores and Source Attributions. In *ACM CHI Conference on Human Factors in Computing Systems*.

Ekin, S. (2023). Prompt engineering for ChatGPT: a quick guide to techniques, tips, and best practices. *Authorea Preprints*.

Gamieldien, Y., Case, J. M., & Katz, A. (2023). Advancing qualitative analysis: An exploration of the potential of generative AI and NLP in thematic coding. *Available at SSRN 4487768*.

Gao, J., Guo, Y., Lim, G., Zhang, T., Zhang, Z., Li, T. J.-J., & Perrault, S. T. (2024). CollabCoder: a lower-barrier, rigorous workflow for inductive collaborative qualitative analysis with large language models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (pp. 1–29)

Gao, A. (2023). Prompt engineering for large language models. *Available at SSRN 4504303*.

Gilardi F, Alizadeh M, Kubli M (2023) ChatGPT outperforms crowd workers for text-annotation tasks. Proc Natl Acad Sci 120(30):e2305016120

Giray L (2023) Prompt engineering with ChatGPT: a guide for academic writers. Ann Biomed Eng 51(12):2629–2633

Goyanes, M., Lopezosa, C., & Jordá, B. (2025). Thematic analysis of interview data with ChatGPT: Designing and testing a reliable research protocol for qualitative research. *Quality & Quantity*, 1–20.

Hendrycks D (2024) Introduction to AI Safety. Taylor & Francis, Ethics and Society (**9781032798028**)

Hou, C., Zhu, G., Zheng, J., Zhang, L., Huang, X., Zhong, T., Li, S., Du, H., & Ker, C. L. (2024). Prompt-based and Fine-tuned GPT Models for Context-Dependent and-Independent Deductive Coding in Social Annotation. In *Proceedings of the 14th Learning Analytics and Knowledge Conference* (pp. 518–528)

Khalid, M. T., & Witmer, A. P. (2025). Prompt Engineering for Large Language Model-assisted Inductive Thematic Analysis. *arXiv preprint* arXiv:2503.22978.

Khan, A. H., Kegalle, H., D'Silva, R., Watt, N., Whelan-Shamy, D., Ghahremanlou, L., & Magee, L. (2024). Automating Thematic Analysis: How LLMs Analyse Controversial Topics. *arXiv preprint* arXiv:2405.06919.

Lee VV, van der Lubbe SC, Goh LH, Valderas JM (2024) Harnessing ChatGPT for thematic analysis: are we ready? J Med Internet Res 26:e54974. https://doi.org/10.2196/54974

Lee, H. P., Sarkar, A., Tankelevitch, L., Drosos, I., Rintel, S., Banks, R., & Wilson, N. (2025). The impact of generative AI on critical thinking: Self-reported reductions in cognitive effort and confidence effects from a survey of knowledge workers. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (pp. 1–22).

Maguire, M., & Delahunt, B. (2017). Doing a thematic analysis: A practical, step-by-step guide for learning and teaching scholars. *All Ireland journal of higher education*, *9*(3).

Mannstadt I, Goodman SM, Rajan M, Young SR, Wang F, Navarro-Millán I, Mehta B (2024) A Novel Approach for Mixed-Methods Research Using Large Language Models: A Report Using Patients' Perspectives on Barriers to Arthroplasty. ACR Open Rheumatology 6(6):375–379

Mansourian Y (2008) Exploratory nature of, and uncertainty tolerance in, qualitative research. New Libr World 109(5/6):273–286

Mathis WS, Zhao S, Pratt N, Weleff J, De Paoli S (2024) Inductive thematic analysis of healthcare qualitative interviews using open-source large language models: How does it compare to traditional methods? Comput Methods Programs Biomed 255:108356

McHugh ML (2012) Interrater reliability: The kappa statistic. Biochem Med 22(3):276–282

Meng, H., Yang, Y., Li, Y., Lee, J., & Lee, Y.-C. (2024). Exploring the Potential of Human-LLM Synergy in Advancing Qualitative Analysis- A Case Study on Mental-Illness Stigma. *arXiv preprint*. arXiv:2405.05758.

Miles, M. B., Huberman, A. M., & Saldaña, J. (2014). *Qualitative data analysis: A methods sourcebook*. (3rd ed.). Sage.

Motoki F, Pinho Neto V, Rodrigues V (2024) More human than human: measuring ChatGPT political bias. Public Choice 198(1):3–23

Nguyen-Trung, K. (2025). ChatGPT in Thematic Analysis: Can AI become a research assistant in qualitative research?. *Quality & Quantity*, 1–34.

Noring, C., Jain, A., Fernandez, M., Mutlu, A., & Jaokar, A. (2024). *AI assisted programming for web and machine learning*. Packt Publishing Ltd.

Patil R, Gudivada V (2024) A review of current trends, techniques, and challenges in large language models (LLMs). Appl Sci 14(5):2074

Sedkaoui, S., & Benaichouba, R. (2024). Generative AI as a transformative force for innovation: a review of opportunities, applications and challenges. *European Journal of Innovation Management*.

Shin, D. (2025). *Debiasing AI: Rethinking the intersection of innovation and sustainability*. Routledge.

Sinha, R., Solola, I., Nguyen, H., Swanson, H., & Lawrence, L. (2024). The Role of Generative AI in Qualitative Research: GPT-4's Contributions to a Grounded Theory Analysis. In *Proceedings of the Symposium on Learning, Design and Technology* (pp. 17–25).

Suter, V., & Meckel, M. (2024). Using GPT-4 for Text Analysis: Insights from English and German Language News Classification Tasks. In *Proceedings of the International of International AAAI Conference on Web and Social Media*.

Tai RH, Bentley LR, Xia X, Sitt JM, Fankhauser SC, Chicas-Mosier AM, Monteith BG (2024) An Examination of the Use of Large Language Models to Aid Analysis of Textual Data. Int J Qual Methods 23:1–14. https://doi.org/10.1177/16094069241231168

Terry G, Hayfield N, Clarke V, Braun V (2017) Thematic analysis. The SAGE Handbook Qualitat Res Psychol 2(17–37):25

Thomas DR (2006) A general inductive approach for analyzing qualitative evaluation data. Am J Eval 27(2):237–246

Turobov, A., Coyle, D., & Harding, V. (2024). Using ChatGPT for thematic analysis. *arXiv preprint*. arXiv:2405.08828.

Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D (2022) Chain-of-thought prompting elicits reasoning in large language models. Adv Neural Inf Process Syst 35:24824–24837

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint* arXiv:2302.11382.

Xiao, Z., Yuan, X., Liao, Q. V., Abdelghani, R., & Oudeyer, P.-Y. (2023). Supporting qualitative analysis with large language models: Combining codebook with GPT-3 for deductive coding. In *Companion proceedings of the 28th international conference on intelligent user interfaces* (pp. 75–78).

Yan, L., Echeverria, V., Fernandez-Nieto, G. M., Jin, Y., Swiecki, Z., Zhao, L., Gašević, D., & Martinez-Maldonado, R. (2024). Human-AI Collaboration in Thematic Analysis using ChatGPT: A User Study and Design Recommendations. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (pp. 1–7).

Zhang H, Wu C, Xie J, Lyu Y, Cai J, Carroll JM (2025) Harnessing the power of AI in qualitative research: exploring, using and redesigning ChatGPT. Comput Hum Behav Artif Hum 4:100144

Zhang, H., Wu, C., Xie, J., Rubino, F., Graver, S., Kim, C., Carroll, J. M., & Cai, J. (2024). When Qualitative Research Meets Large Language Model- Exploring the Potential of QualiGPT as a Tool for Qualitative Coding. *arXiv preprint*.14925.