Preliminary Study: Conducting Qualitative Thematic Analysis with Large Language Models and RAG Implementation

Natalie A. Barnett

MSc in Applied Information and Data Science

Lecturer: Rabea Krings

Co-lecturer: Diego Antognini

Lucerne 31 May, 2024

Lucerne University of Applied Sciences and Arts

1. Background

In the field of data science, much of the research conducted depends on numerical figures to analyse, optimise or predict with large amounts of data. It is a multidisciplinary approach historically built on the principles of mathematics and statistics to extract meaningful insights (Provost & Fawcett, 2013). In recent years a subfield of data science, Natural Language Processing (NLP), has made great strides in moving beyond simple statistical pattern recognition to assimilating human perception through advanced neural networks called Large Language Models (LLMs). These models, trained on immense amounts of data, are a form of generative Artificial Intelligence (AI) that have revolutionised data science by demonstrating the capacity to provide solutions to complex, ambiguous tasks (Brown et al., 2020). As AI is incorporated into increasingly more workflows beyond the traditional quantitative data applications, the question arises if AI can also be integrated into, or even fully replace, the manual labour of qualitative research as well.

Qualitative research primarily tackles non-numeric data, such as those from observational studies or interviews, and depends solely on human interpretation and philosophical frameworks to identify patterns or trends. A strong understanding of human language, perceptions and behaviour is essential to gather meaningful insights from textual data. For this reason, skilled researchers are typically recruited for the task, who must invest large amounts of time and resources to read, code and interpret text to identify themes (Creswell & Poth, 2016). LLMs on the other hand, can process vast amounts of textual data and are rapidly demonstrating the amazing ability to statistically perform what we, as humans, do semantically (Floridi, 2023). ChatGPT, an LLM within the field of AI, has recently grown in popularity due to its ability to converse and answer questions in a human-like manner while simultaneously completing some tasks more accurately and efficiently than humans (Welivita & Pu, 2024; Luo et al., 2024; Wei et al., 2024). Because of this, it is reasonable to propose that LLMs, like ChatGPT, can reduce the human labour involved with qualitative analysis through its deep language comprehension skills.

This research project aims therefore to demonstrate the ability of LLMs to perform qualitative analysis and identify the manner in which to utilise these models for the desired results effectively and efficiently.

2. Topic Definition

Thematic analysis, a foundational aspect of qualitative research, is an approach to analysing data where patterns (themes) are not only identified, but also interpreted into meaningful insights from topic-specific texts. Often used with interview or focus group data, thematic analysis utilises a flexible approach allowing the researcher to provide their own perspective and expertise into the analysis by building a purposeful story from the data (Braun & Clarke, 2019). The process of identifying themes consists of identifying representative data extracts, known as codes, in the text and then systematically aggregating them into broader themes . There are two primary ways to identify themes within data: an inductive 'bottom-up' approach and a deductive 'top-down' approach. Deductive reasoning is typically theory driven in that specific instances are found in the data to support an existing hypothesis, while inductive reasoning is more exploratory in nature and develops themes from the data (Braun & Clarke, 2006). The inductive approach, the focus for this project, does not rely on previous research, and thus is extremely labour intensive as it requires a strong familiarity with the data before performing the analysis.

Due to the intense time requirements of inductive thematic analysis, specialised software is often used to assist researchers by simplifying data management and offering collaborative work platforms. Some tools even help identify codes though artificial intelligence and then combine them into AI generated themes. Regardless of these advancements, the current tools on the market are still limited in their capabilities and require significant human supervision throughout the process (refer to Table 1 in appendix). In addition, these tools often come with hefty subscription costs and a steep learning curve to efficiently operate. Therefore, there is still a strong interest in the application of LLMs for a more individualised and simplified approach to inductive thematic analysis.

For this project, we aim to automate and streamline the process by capitalising on the depth of knowledge in LLMs to complete an inductive thematic analysis using real focus group data. We will use two different language models to identify and translate potential themes within the transcript data according to the famous six step framework laid out by Braun and Clarke (2006). The framework consists of: "(1) familiarising yourself with your data; (2) generating initial codes; (3) searching for themes; (4) reviewing themes; (5) defining and naming themes; (6) producing the report" (see Table 3). Due to the simplicity and clarity of the methodology, we

believe it is possible to translate these steps into actionable commands for the LLM to produce a credible analysis.

The focus group data for the analysis explores and discusses with medicine doctors their experience of wearing glucose sensors used by patients with diabetes. Ideally, we hope to identify, through inductive thematic analysis, how self-tracking with a glucose sensor influences residents' awareness, appreciation and understanding of glucose metabolism in diabetics. The goal is to explore the necessary methods or required architecture to complete a viable thematic analysis with LLMs and then develop a formal procedure that can be applied to research of diverse topics. Additionally, we want to understand how the analysis of the two LLMs compare in performance and usability.

2.1 Research Questions

2.11 How do prompts for LLMs need to be formulated and how do they need to be used sequentially to achieve the best results for thematic analysis? Since the rise of AI, the emerging field of prompt engineering has gained popularity due to the necessity of well-crafted prompts. Prompts "are instructions given to an LLM to enforce rules, automate processes, and ensure specific qualities (and quantities) of generated output" (White et al., 2023). Although LLMs have a strong grasp of intention in requests, it does require some trial and error in the wording to receive the desired results. Additionally, slight variations in the prompt can produce vastly different responses in terms of adequacy. To structure the prompts and explore the output evolution, we will be using three well-known prompt engineering techniques: zero-shot, few-shot and chain-of-thought (CoT) prompting.

Furthermore, not only do prompt formats need to be considered, but also the sequence in which they are inputted into the LLM. In the case of CoT, in which tasks are broken down into smaller steps, the order in which intermediary steps are introduced may result in different reasoning patterns used by the model. This will need to be explored to discover the necessary sequence to achieve quality outputs.

2.12 How does the addition of complementary, relevant data, in the form of a Retrieval-Augmented Generation architecture, affect the prompt outputs? Retrieval-Augmented Generation (RAG) is a way to introduce internal or supplementary data to a pre-trained language model before initiating a task. Rather than re-training a computationally

expensive LLM, we can build a database with the additional data, and then retrieve the relevant information before sending the specific model request (Lweis et al., 2020). Essentially, this will provide more context to our inputs and ensure the most up-to-date, necessary information is included with our carefully crafted prompts.

Once the RAG architecture is in place, we will run the same prompts into the new pipeline and identify the extent to which the outputs change. We also must evaluate the RAG performance itself and ensure relevant information is accurately retrieved from the database.

2.13 How and to what extent do the results of different language models differ in terms of quality? Of the two LLMs tested, we want to evaluate the results and determine which model produces the best thematic analysis. Braun and Clarke provide a "15-point checklist of criteria for good thematic analysis", as seen in Table 4, that will be used as the standard when judging the LLM outputs (2006, p.96). This evaluation will be conducted with subject matter experts using a rating system to provide a credible assessment of how the LLM performs in each criterion. Simply, the LLM with better ratings will be deemed better at thematic analysis. Given that qualitative studies are an inherently subjective task, without one ground-truth answer, we can assume that a human evaluation of the results provides meaningful insight into LLM performance quality.

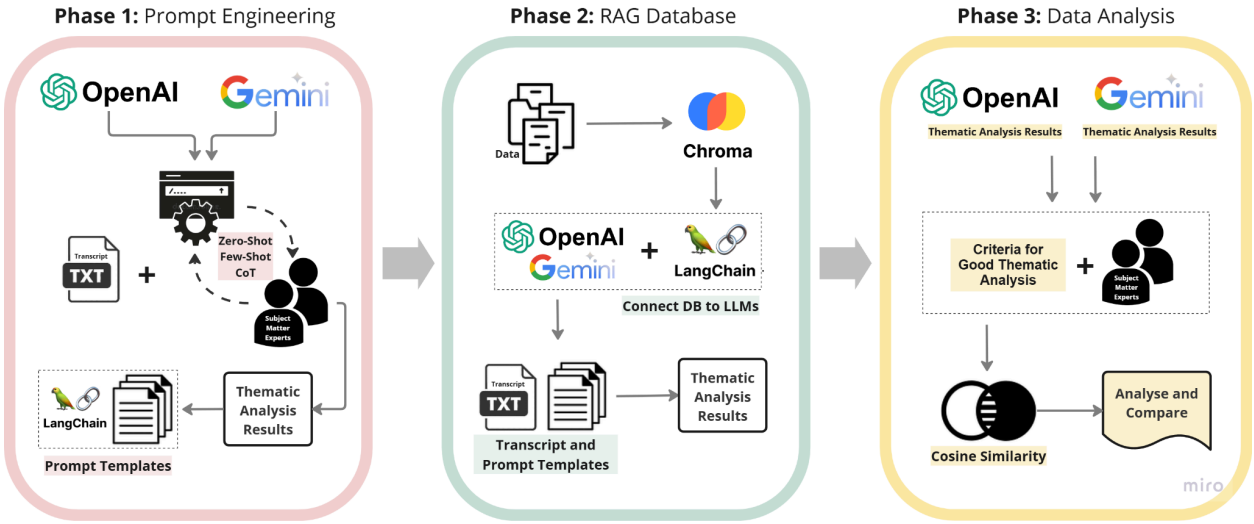
Additionally, we will numerically calculate the similarity between both LLM results and produce a visual representation of how the analyses compare. LLMs vary in size and context window lengths which will likely have an effect on the output quality. This comparison can provide insight into the effect of LLM specifications on thematic results.

3. Methodology

The project will be conducted in three phases as shown in Figure 1. Phase one will be an exploration of prompt engineering techniques to formulate effective prompts for completing the thematic analysis steps. During this phase, we will collaborate with qualitative research experts to ensure the prompt inputs and outputs are in line with current standards. The next phase includes building and incorporating a RAG database to the LLM workflow and evaluating its performance on the same prompts developed in phase one. Finally phase three is the evaluation phase where results from both phases will be analysed with the subject matter experts and compared using cosine similarity

Figure 1

Project Methodology



3.1 Choosing the Large Language Models

The decision of which LLMs to use in the study depends on LLM benchmark performance, context window size, cost and privacy concerns, from highest to lowest priority. The benchmark we are using to assess LLM performance quality is Chatbot Arena, an open platform for evaluating LLMs based on human preferences. Chatbot Arena crowdsources a diverse user base to vote on LLM output preference using pairwise comparison and then uses this information to estimate the ranking of models (Chiang et al., 2024). A higher ranking indicates a model produces preferable responses according to real human feedback. Context window refers to the maximum sequence that can be processed by the LLM at one time (Hughes, 2024). This means that the larger the context window, the more words can be inputted into the prompt at once which the model will remember and retain. Cost is also considered because the majority of the newest, most powerful models incur a cost for every request sent and response received from the LLM. Finally, privacy is also taken into consideration because the LLM providers are known to retain and train their models with user inputs. In regards to qualitative research, this poses a risk because it is often essential to maintain the confidentiality of the study participants and their information. With this potential breach of confidentiality, researchers may be limited in their use of LLMs, unless privacy is guaranteed and secure. With all of these

considerations in mind, we compare the most common and well known LLMs in Table 2 including GPT, Gemini, Claude and Mistral.

Table 2

Comparison of Large Language Models

LLM Comparison						
Provider	**Privacy Policies for Inputs/Outputs**	**API Model Name**	**Input Cost (per 1M Tokens)**	**Output Cost (per 1M Tokens)**	**Context Window (Tokens)**	**Chatbot Arena Ranking**
OpenAI	When using the browser interface, content is collected and may be used to train new models. There is an option to opt out by filling in a form. When LLM is accessed via API, content is not used to improve models and services. API data may be retained for up to 30 days.	gpt-3.5-turbo-0125	$0.50	$1.50	16,385	42
		gpt-4-turbo-2024-04-09	$10.00	$30.00	128,000	2
		gpt-4o-2024-05-13	$5.00	$15.00	128,000	1
Anthropic	Input and output content may be processed in aggregated or de-identified forms for training AI models. Prompts and outputs are automatically deleted on the backend within 90 days of receipt or generation.	claude-3-haiku-20240307	$0.25	$1.25	200,000	17
		claude-3-sonnet-20240229	$3.00	$15.00	200,000	12
		claude-3-opus-20240229	$15.00	$75.00	200,000	6
Google	With paid services, prompts or responses will not be used to improve products. Unpaid services are used to help with quality and improve products. Human reviewers may read and process anonymous conversations, which are retained for up to 3 years. Under certain privacy laws, the processing of data can be objected by creating a request.	gemini-1.5-Flash-API-0514	$0.70	$2.10	1,000,000	9
		gemini-1.5-Pro-API-0409-Preview	Free	Free	32,000	4
		gemini-1.5-Pro-API-0514	$7	$21	1,000,000	2
Mistral AI	Content is only used to improve models if opted into the Mistral AI Training Data option. Prompts and outputs are retained for 30 rolling days to monitor abuse.	mistral-large	$4	$12	32,000	21
		mistral-medium	$2.7	$8.1	32,000	25

Note. Privacy and pricing information was collected from the provider website and was accessed on 30.05.24
Note. Chatbot Arena ranking was collected on 30.05.24 (https://chat.lmsys.org/?leaderboard)

The new GPT-4o model is ranked number one in human preference in the Chatbot Arena followed by a tie in second place between GPT-4 Turbo and Gemini 1.5 Pro. In regards to context windows however, Google is the clear winner with around one million tokens for both the Gemini Pro and Flash models. As for pricing, cost is calculated per one million tokens and shows all models are priced comparably, with the exception of Claude 3 Opus being slightly more expensive for both input and output costs. The data privacy policies, which here we focus only on the processing of user inputs and outputs, reveals three of the four providers exploit user content to further train their language models. This means information fed into prompts is anonymised to a certain degree, retained and used in future model updates. Only Mistral guarantees that input and output data is private to only the user and not to improve their services. However, inputs and outputs are stored by all providers ranging from 30 days to as long as 3 years to monitor abuse among other purposes. Given that privacy policies are heavily governed by local regulations and that these policies are constantly evolving, data privacy considerations will vary greatly depending on location. Because we have consent from participants to share the focus group data, we will only focus on performance ranking, context window and cost for model selection and deprioritize privacy. Nevertheless, we recognize privacy is a primary concern in most qualitative research studies and want to address our reasoning for deprioritizing privacy considerations in our project.

With these considerations in mind, we will perform the study with Gemini 1.5 Pro (paid model) and GPT-4o, because they offer the best balance of large context windows, high performance ranking and low cost. Plus, Gemini 1.5 Pro and GPT-4o offer latest advancements in the AI field which will be needed to perform the complex reasoning tasks in a thematic analysis.

3.2 Phase 1: Prompt Engineering

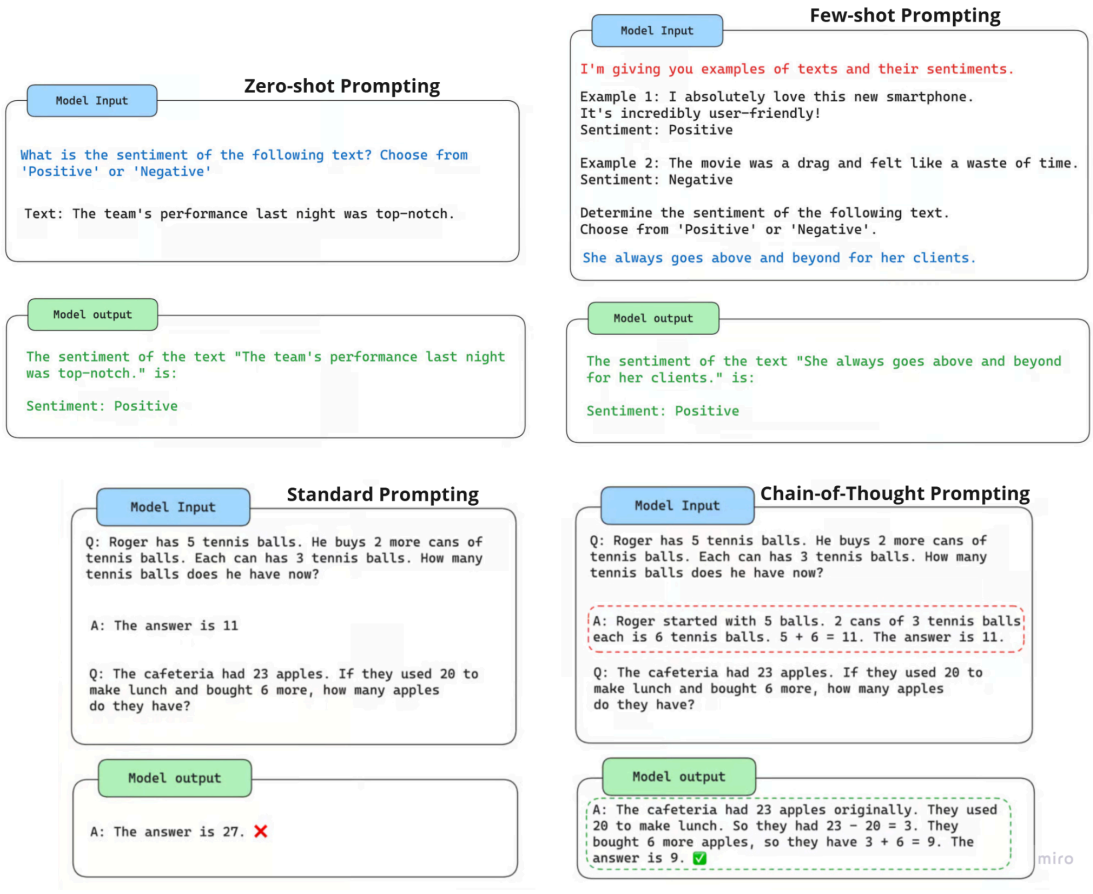
If prompts are the set of instructions provided to the LLM, then prompt engineering is the means in which LLMs are programmed by the given prompts (White et al., 2023). Prompt engineering is a growing field where new techniques are constantly being discovered and developed to curate LLMs towards a specific output. However, with this project, we will be focusing on only three well-established prompt engineering techniques to understand the baseline capabilities of GPT-o4 and Gemini 1.5 Pro, as well as simplify prompts to be easily

interpretable for qualitative researchers seeking to incorporate AI into their workflows. The three methods we will use are zero-shot, few-shot and chain-of-thought (CoT) prompting.

Zero-shot prompting is the most basic form of prompting in which natural language alone describes the task to be performed. This will be used first as a benchmark to understand the model's baseline capabilities in handling abstract requests. Next, we will implement few-shot prompting, popularised by Brown et al. (2020), where a model is initialised with a description and multiple examples of the relevant task. Few-shot performance is often much higher due to the LLM's ability to learn in the context of requests. Finally, we will introduce CoT prompting where a problem is decomposed to intermediate steps which mimic a human thought process when solving a complicated task (Wei et al., 2023). CoT prompting is an effective method for enhancing reasoning within LLMs but has not been thoroughly tested in the context of qualitative analysis. Figure 2 provides an example of each prompting technique.

Figure 2

Examples of Prompting Techniques



Note. From "A Guide to Prompt Engineering", by A. Pachaar, 2023, (https://mlspring.beehiiv.com/p/guide-prompt-engineering)

The prompts will be developed in collaboration with qualitative research experts to ensure the inputs fulfil the step-by-step process laid out by Braun and Clarke (2006), as seen in Table 3. We will modify these steps into commands interpretable by the LLM, format the commands data according to a prompting technique and send the developed prompt to the LLM for a response. The prompt development will be an iterative process whereby the initial prompt will be changed and improved per expert feedback until a desirable output is achieved. Ultimately, we should end up with three finalised prompts for each LLM, one for each chosen prompting technique, for a total of six prompts at the end of Phase 1. The finalised prompts will be transformed into langchain prompt templates which will automate multi-step prompts into one line of code. The templates will save time in Phase 2 by simplifying the commands needed to re-execute the prompts.

Table 3

Step-by-step Instructions of a Thematic Analysis

Phase	Description of the process
1. Familiarizing yourself with your data:	Transcribing data (if necessary), reading and re-reading the data, noting down initial ideas.
2. Generating initial codes:	Coding interesting features of the data in a systematic fashion across the entire data set, collating data relevant to each code.
3. Searching for themes:	Collating codes into potential themes, gathering all data relevant to each potential theme.
4. Reviewing themes:	Checking if the themes work in relation to the coded extracts (Level 1) and the entire data set (Level 2), generating a thematic 'map' of the analysis.
5. Defining and naming themes:	Ongoing analysis to refine the specifics of each theme, and the overall story the analysis tells, generating clear definitions and names for each theme.
6. Producing the report:	The final opportunity for analysis. Selection of vivid, compelling extract examples, final analysis of selected extracts, relating back of the analysis to the research question and literature, producing a scholarly report of the analysis.

Note. From "Using Thematic Analysis in Psychology" by V. Braun and V. Clarke, 2006, *Qualitative Research in Psychology, 3*, p. 87.

3.3 Phase 2: RAG Implementation

After completing the prompt analysis, we will introduce a RAG architecture into the pipeline. Initially developed by Meta AI researchers, RAG introduces an intermediary step between the input sequence (prompt) and the output sequence (response) of the LLM. Rather

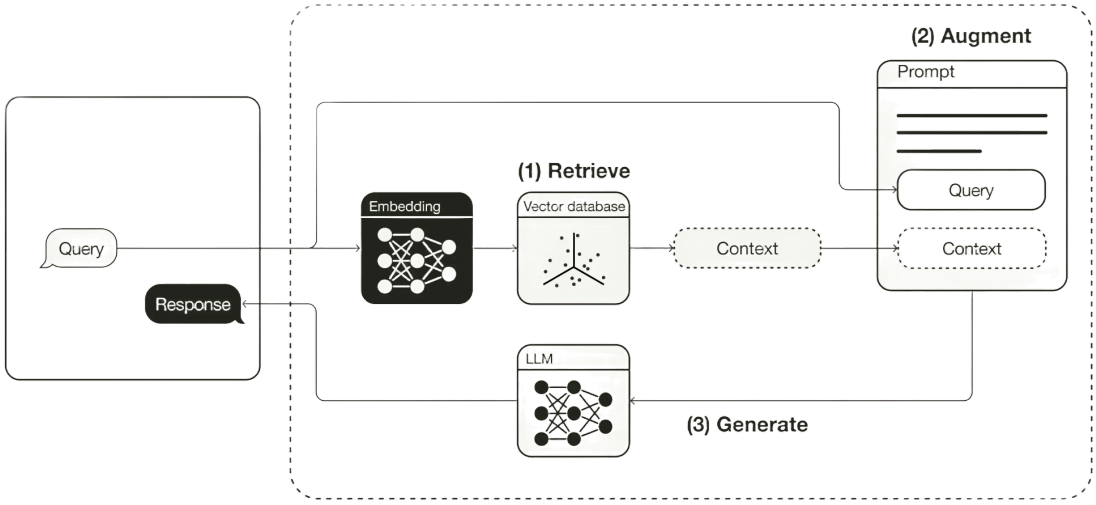
than the input being sent directly to the LLM as in phase one, RAG first uses the input to retrieve relevant, supplemental data from a separate database and enriches the prompt with additional context before sending for a response (Meta, 2020). The architecture utilises both parametric knowledge from the LLM and external nonparametric knowledge from the RAG database to improve performance.

Because LLMs often have knowledge cutoff dates and it is unknown what data is used to train closed models, a RAG architecture provides a work-around to cater models to a specific task. In our case, we will build a RAG database centred around the focus group topic to ensure the model grasps the meaning of the text before performing the thematic analysis. The benefit of RAG implementation is that studies show it improves responses in knowledge-intensive tasks and reduces hallucination (Lewis et al., 2020). Due to the complex nature of thematic analysis, the supplemental data may aid the model in interpreting the focus group data and produce better results.

To build the RAG database, we first need to ingest the external corpus data with document loaders and split the data into smaller chunks. Document splitting is useful in that it breaks up long documents into smaller sections to be able to fit within an LLM's context window (Bennion et al., n.d.). The documents can also be overlapped in the splitting process to assure the chunks retain context. Once split, the documents will be encoded and embedded to be easily retrievable by the RAG. The embedding process numerically represents the documents as vectors which can then be stored in an external vector database. The database is made available to the LLM through retrievers, as shown in Figure 3, which embed the prompt into the same vector space as the vector database. A similarity search between the prompt and external data will pull the top closest objects from the vector database and then both the prompt and new context will be sent to the LLM for a response (Monigatti, 2023).

Figure 3

Retrieval-Augmented Generation Workflow



Note. From "Retrieval-Augmented Generation (RAG): From Theory to LangChain Implementation", by L. Monigatti, 2023, *Towards Data Science* (https://towardsdatascience.com/retrieval-augmented-generation-rag-from-theory-to-langchain-implementation-4e9bd5f6a4f2).

Before moving forward with the entire thematic analysis after the RAG implementation, it is important that we evaluate the ability of the RAG to retrieve relevant passages, utilise these passages in a faithful way and assess the relative quality of generated content. These aspects can be measured reference-free through the Retrieval Augmented Generation Assessments (RAGAs) introduced by Shahul et. al (2023). To assess with RAGAs, we first need a question-answer data set based on our RAG documents to use in the performance assessment. Building a list of question and answer combinations is very time consuming, but thankfully the RAGAs python package, which we will use for this project, has a synthetic test data generator that creates question-context-answer combinations for us from the uploaded RAG documents. It is with this data set that we will be able to measure the following RAGAs values.

The first metric to be considered in RAGAs is Faithfulness referring to "the idea that the answer should be grounded in the given context" (Shahul et. al, 2023, p. 3). This is important to avoid LLM hallucinations and ensure the retrieved documents from the RAG act as justification for the answer. A faithfulness score is calculated by using dedicated prompts to identify the number of statements that support the chosen questions (V) divided by the total number of statements (S). The formula is:

$$F = \frac{|V|}{|S|}$$

The resulting score is a scaled range between 0 and 1 where a higher value indicates better faithfulness.

Next is Answer Relevance which identifies if the generated answer actually addresses the original question. We can test this by asking the LLM to generate a question based on any given output. Then by embedding the resulting question, we can calculate the cosine similarity between the original question and the generated question with the following formula:

$$\text{AR} = \frac{1}{n} \sum_{i=1}^{n} \text{sim}(q, q_i)$$

Higher scores indicate more relevance whereas a lower score indicates answers with incomplete or redundant information.

Last is Context Relevance, a concept that shows the extent to which the retrieved content contains information needed to answer the questions and penalises redundant information. This score is calculated by extracting relevant sentences to one prompted question. Then, the number of extracted sentences is divided by the total number of sentences to determine their relevance score. The equation for this measure is:

$$\text{CR} = \frac{\text{number of extracted sentences}}{\text{total number of sentences in } c(q)}$$

The resulting score is a scaled range between 0 and 1 where a higher value indicates higher relevancy.

All metrics, laid out by Shahul et al (2023), signal better execution with a higher score and worse execution with a lower score. As stated before, this is a reference-free evaluation metric that can be used as an indicator of relative RAG performance, not a definitive source of performance quality. Through fine-tuning of parameters, we will ensure that the RAG architecture is properly working before moving on in the analysis.

Once the RAG is fine-tuned, both LLMs will be connected to the resulting vector database, and the previously developed prompt templates will be executed within the new RAG architecture. The resulting thematic analyses from both phases one and two will then be ready for evaluation and analysis.

3.4 Phase 3: Data Analysis and Evaluation Metrics

At this point in the project, we will now have a total of 12 completed thematic analyses. In collaboration with our subject matter experts, we will determine the prompting technique that produces the best results for each model, assess the effects of a RAG architecture on the outputs and finally, decide on the LLM that produces the best thematic analysis.

Where needed throughout the evaluation process, we will also calculate cosine similarity between two or more thematic analyses and visually or numerically compare the results. This will not be an evaluation metric to decide which outputs are better, but rather a way to track similarities or differences between the LLM outputs. Below we will explain in more detail how cosine similarity and our other evaluation metrics operate and where they will be used in the context of our project.

3.41 How do we compare thematic analysis results with cosine similarity? Cosine similarity is a common metric used in NLP to measure the semantic agreement between two pieces of text. This is done by first transforming the text into vector representations through embedding, a technique which numerically captures the nuanced essence of words relationally and contextually. The vectors can then be represented in a multi-dimensional space allowing for the ability to measure the cosine of the angle, or cosine similarity, between two vectors. The similarity score ranges from -1 to 1 with 1 denoting perfect similarity and -1 complete dissimilarity (Grønne, 2022).

In the context of our project, we can use the cosine similarity between two LLM outputs to capture semantic overlap. We will calculate the similarity scores of all three prompting techniques for each LLM as well as compare the semantic agreement between prompts with and without the RAG architecture in place. Finally, we will also measure and visually represent the cosine similarity between the two LLMs and their outputs using a clustering algorithm. Incorporating cosine similarity throughout the project provides an intuitive mode of comparison as well as enhances our ability to understand the language models.

3.42 How do we evaluate the LLM outputs to determine the best thematic analysis? Evaluating the quality of thematic analysis is an inherently subjective task, and there are no correct or incorrect answers to any one analysis. Without ground-truth labels for evaluation, we will use Braun and Clarke's list of criteria for a good thematic analysis (see Table 4) as a guide to score the analysis outputs. We will use a five point Likert scale to rate the degree to which the

thematic analysis meets each of the criteria in the 15-point checklist. An example of how the evaluation will look can be seen in Figure 4 where each option corresponds to a numerical value ranging from 1 (strongly disagree) to 5 (strongly agree). Based on their expertise, our qualitative researchers will rate the outputs on their level of agreement to each statement. In the end, we will be able to summarise the results for all twelve thematic analyses and identify the prompting techniques that resulted in the best results, define the effects of a RAG architecture on output quality and reveal which LLM is best suited for thematic analysis.

Table 4

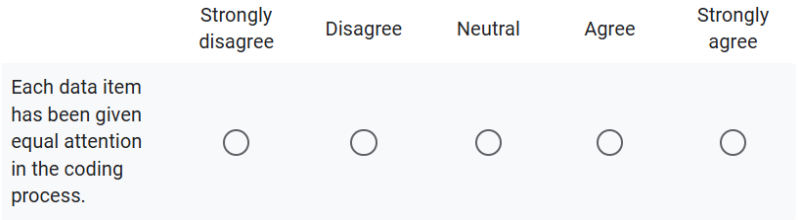
A 15-point checklist of criteria for a good thematic analysis

Process	No.	Criteria
Transcription	1	The data have been transcribed to an appropriate level of detail, and the transcripts have been checked against the tapes for 'accuracy'.
Coding	2	Each data item has been given equal attention in the coding process.
	3	Themes have not been generated from a few vivid examples (an anecdotal approach), but instead the coding process has been thorough, inclusive and comprehensive.
	4	All relevant extracts for all each theme have been collated.
	5	Themes have been checked against each other and back to the original data set.
	6	Themes are internally coherent, consistent, and distinctive.
Analysis	7	Data have been analysed – interpreted, made sense of – rather than just paraphrased or described.
	8	Analysis and data match each other – the extracts illustrate the analytic claims.
	9	Analysis tells a convincing and well-organized story about the data and topic.
	10	A good balance between analytic narrative and illustrative extracts is provided.
Overall	11	Enough time has been allocated to complete all phases of the analysis adequately, without rushing a phase or giving it a once-over-lightly.
Written report	12	The assumptions about, and specific approach to, thematic analysis are clearly explicated.
	13	There is a good fit between what you claim you do, and what you show you have done – ie, described method and reported analysis are consistent.
	14	The language and concepts used in the report are consistent with the epistemological position of the analysis.
	15	The researcher is positioned as *active* in the research process; themes do not just 'emerge'.

Note. From "Using Thematic Analysis in Psychology" by V. Braun and V. Clarke, 2006, *Qualitative Research in Psychology, 3*, p. 96.

Figure 4

Example of Likert Scale for Evaluation



3.44 How do we validate that the resulting procedure can perform an accurate thematic analysis? Upon completing phase 3 of the project, we ideally will be able to recommend a formal procedure of how to utilise LLMs in thematic analysis. We can validate the efficacy of the new procedure by testing the steps on a new qualitative data set and comparing the generated themes with the human generated themes of the original study. By embedding the themes and calculating cosine similarity, we can visually represent the level of agreement between the original results of the study and the new approach using the LLM. A high level of agreement would indicate that the new LLM procedure results in themes that are generally accurate and acceptable by the research community.

4. Underlying Data

4.1 Transcript Data

The focus group data set was collected and transcribed by Gurpreet Anand of the University of Bern and contains approximately 38,000 words in total. The initial aim of the study included exploring and describing the experiences of medicine doctors after wearing glucose sensors for 14 days. The population of doctors includes 22 residents working in internal medicine at the Zollikerberg hospital and the chosen glucose sensor for monitoring glucose levels was the Freestyle libre (FSL). The research questions for this study were:

1. *How can self-tracking with a glucose sensor influence residents' understanding of glucose metabolism?*
2. *How can self-tracking with a glucose sensor improve residents' awareness, appreciation and understanding of patients with diabetes?*

The study participants granted consent to use the collected focus group data for research. Furthermore, the focus groups were originally conducted in German but have been translated into English using a DeepL pro account.

4.2 RAG Data

The vector database for the RAG architecture will include all supporting documents and research involved in the original study by Gurpreet Anand. Additionally, operating manuals as well as informational brochures from the Freestyle libre corporate website will be included into the RAG database. The files for this can be found here from the corporate website:

[https://www.freestyle.abbott/ch-de/hilfe/videos-und-downloads.html]. All files will be translated into English as needed using a DeepL Pro account.

5. Technology

All work involved in this project will be conducted in Python through a Google Colab Pro account. Colab offers multiple benefits including access to high-memory VMs and longer runtimes. The LLMs, GPT-4o and Gemini 1.5 Pro, will be accessed by an Application Programming Interface (API) via python script. The APIs allow us to seamlessly integrate the LLM with other tools within Python as well as provide more control over the data handling and manipulation.

Within Python, we will be working primarily with the LangChain package. LangChain provides a framework for developing LLM applications featuring chains, agents and flexible integration of APIs. Chains, call sequences used to create complex workflows, and agents, components used in dynamic decision-making processes, will be used to develop the prompt templates and integrate the RAG architecture for both LLMs. Langchain also offers document loaders and text splitters to prepare the RAG data for embedding.

In order to embed text, we will be using the `Linq-Embed-Mistral` embedding model from Mistral available on Hugging Face. It is open-sourced, free and currently ranks third on the Massive Text Embedding Benchmark (MTEB) Leaderboard[1]. The vector database to store the embedded text will be hosted through Chroma, an open-source embedding database. Chroma is free to use and easily integrates with LangChain and various embedding models. To evaluate RAG performance, we will use the ragas package which facilitates the calculation of RAGAs scores.

6. Annotated Disposition

Below is a suggested disposition for the final dissertation. Some sections provide a short explanation of the contents unless otherwise self-explanatory.

1. Introduction - *introduce the problem and why it is important*
    1.1. Research Questions

[1] From https://huggingface.co/spaces/mteb/leaderboard Accessed on 31.05.24

2. Background and Related Work - *Explain the data used in the study and share information* about related work
3. Methodology- *Description of project phases*
    3.1. Large Language Models - *Discuss how the models were chosen and the interface (API)*
    3.2. Prompting Techniques - *Explanation of the process of creating prompts, the chosen prompting techniques and the collaboration with subject matter experts*
        3.2.1. Zero-shot
        3.2.2. Few-shot
        3.2.3. Chain-of-Thought
    3.3. RAG Architecture - *Explanation of RAG and how it is incorporated*
        3.3.1. RAG Data - *Share information about the data used in the RAG*
4. Results and Evaluation - *Provide results and evaluation criteria for each section*
    4.1. Prompt Engineering
    4.2. RAG Implementation
    4.3. Comparison - *Compare the results between the two LLMs with cosine similarity and subject matter expert opinions*
5. Conclusion - *Answer the research questions*
6. Recommendation and Limitations - *Provide a step-by-step guide of how to incorporate LLMs to thematic analysis (if possible). Also, discuss any limitations involved throughout the project (i.e cannot claim the prompts are the best options).*
7. Bibliography
8. Appendix - *dictionary of abbreviations, additional tables, transcript data, etc.*

7. Project Risks

Throughout the project execution, there will be risks that must be managed and mitigated. These risks include cost of execution, prompt development options, hallucination or false responses, biased outputs, RAG tuning and overall timing.

The cost involved with using the LLM APIs can easily exceed a reasonable amount if requests are not properly monitored. Charges are incurred for every input token and every output token within the Open AI and Gemini APIs. While generally the costs per token can be cheaper

than a monthly subscription to a chat interface, such as ChatGPT, they can still add up, especially during the prompting phase where there will be significant experimentation with inputs and responses. To mitigate these costs, we will perform initial testing with the free version of Google Gemini or with GPT-3.5, a significantly cheaper model than the newest GPT-4o model. Once the prompts are close to being finalised, we will run them through the models chosen for this study. This prevents dozens of aimless requests from being sent though expensive models and focuses on only sending the best prompts.

As for the prompting phase, there is no guarantee that we will truly find the "best" prompts for the task at hand. Prompt techniques are constantly evolving and there is no set standard for prompting within the qualitative research space. Additionally, there are an infinite number of ways to word requests and format the LLM outputs. Therefore we will need to rely heavily on our subject matter experts with building the prompts and formatting outputs in a way that matches current practices in the field. We will also refer to previous prompt engineering research to enhance our prompts, such as suggestions laid out by White et al in "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT" (2023).

We also must be attentive to false, hallucinated or biased responses from the LLMs during the prompt testing phase. LLMs are known to generate what seem to be coherent answers but can be entirely made up. RAG architectures often help reduce false or hallucinated responses by providing additional context to the request. This can be further mitigated by providing examples in the prompts, as we will with few-shot prompting, and instructing the LLM to admit when it doesn't know the answer. We can also monitor incorrect answers by asking the model to support proposed themes with supporting quotes from the transcript data.

When it comes to LLM bias however, this is trickier to address because it is the result of the training data used to build the actual model. The training data may contain false or culturally biased information, but the LLM will interpret it as factual and use the information as foundational knowledge. Within the RAG architecture, we can monitor this by ensuring the retrieved context relates to the prompt. Outside of the RAG though, we will need to rely on our subject matter experts to identify potential bias and address it within the thematic analysis.

Finally, we also need to consider the time requirements for using LLMs in research. It is a possibility that "Reading, understanding, and evaluating ChatGPT results uses no less time than

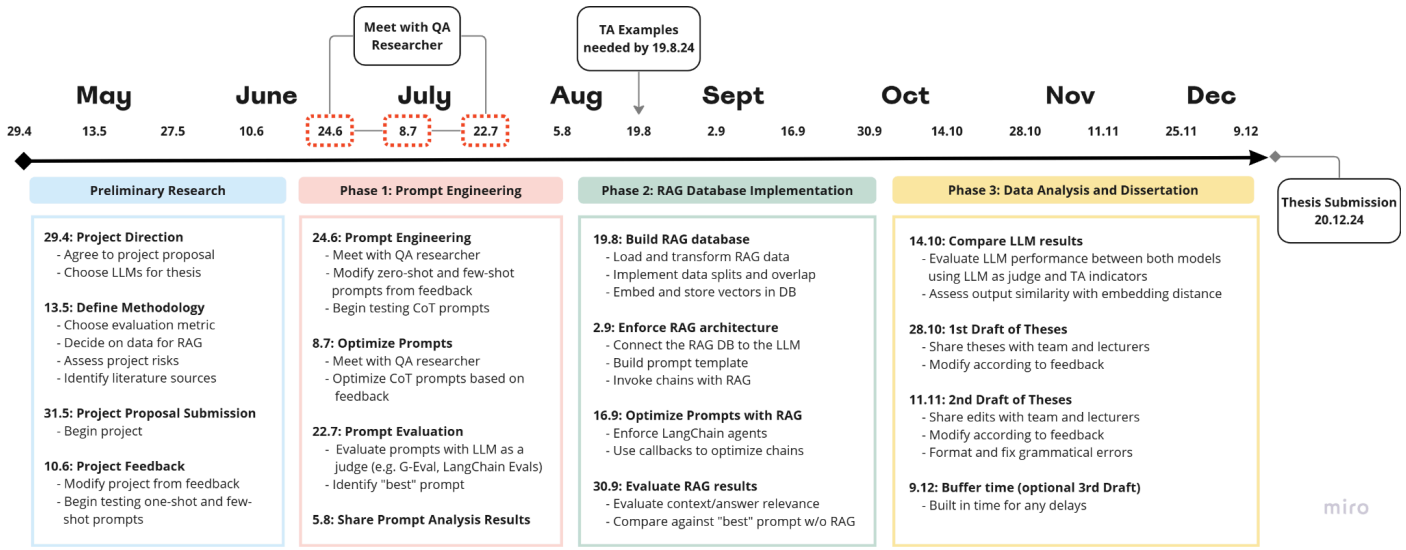
the original workflow" (Zhang et al., 2023). In this case, we must reflect on the benefits of LLMs in qualitative research and to what extent these tools should be used to save time and resources.

8. Work and Research Plan

Figure 5 provides a timeline of work that will be completed until the final submission.

Figure 5

Timeline of Work until December 20 Submission



9. Annotated Sources and Literature

Bennion, J., & Chapman, J. (n.d.). Developing LLM Applications with LangChain. DataCamp. https://app.datacamp.com/learn/courses/developing-llm-applications-with-langchain

➔ *Bennion & Chapman's course on Datacamp is an incredibly useful and practical source of information for building a RAG architecture using Langchain. They walk through the process step by step and provide examples of code to follow.*

Braun, V., & Clarke, V. (2006) Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2), 77-101.

Braun, V., & Clarke, V. (2019) Reflecting on reflexive thematic analysis. Qualitative Research in Sport, Exercise and Health, 11(4), 589-597, DOI: 10.1080/2159676X.2019.1628806

➔ *Braun and Clarke are the forefront researchers in thematic analysis. Both papers written by them will be constantly referred to throughout the project to ground our understanding of the thematic analysis approach.*

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. NeurIPS.

➔ *This is a fundamental source in data science that describes the advancement of LLMs to perform agnostic tasks.*

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J. E., & Stoica, I. (2024). Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. arXiv:2403.04132 [cs.AI]. https://doi.org/10.48550/arXiv.2403.04132

➔ *Chatbot Arena is a recently developed benchmark in LLM evaluation. We will need to refer to this paper and the leaderboard during this project to stay updated on model performance and new developments in the field.*

Creswell, J. W., & Poth, C. N. (2016). Qualitative Inquiry and Research Design: Choosing Among Five Approaches. SAGE Publications.

➔ *This is another source of information for qualitative research methods. We can refer to this book for information about qualitative research topic overviews and explanations of terminology.*

Floridi, L. AI as Agency Without Intelligence: on ChatGPT, Large Language Models, and Other Generative Models. Philos. Technol. 36, 15 (2023). https://doi.org/10.1007/s13347-023-00621-y

➔ *An editorial type of article, Floridi offers a contrary opinion of LLMs and shows their pitfalls in performance.*

Grønne, M. (2022, September 15). AN EXTENSIVE INTRODUCTION TO LATENT SPACES: Introduction to Embedding, Clustering, and Similarity. Towards Data Science. https://towardsdatascience.com/introduction-to-embedding-clustering-and-similarity-11dd80b00061

➔ *Grønne simply explains how to use cosine similarity in NLP applications. We will refer to this article when calculating and creating visuals with the cosine calculation.*

Hughes, C. (2024, January 27). De-Coded: Understanding Context Windows for Transformer Models. Towards Data Science. https://towardsdatascience.com/de-coded-understanding-context-windows-for-transformer-models-abcdef123456

➔ *Hughes explains how context windows work and their affect on LLMs.*

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research. 119:3929-3938. https://proceedings.mlr.press/v119/guu20a.html

➔ *This article from Meta researchers introduced RAG to the data science field and walks through the technical and mathematical reasoning behind its application.*

Luo, X., Rechardt, A., Sun, G., Nejad, K. K., Yáñez, F., Yilmaz, B., Lee, K., Cohen, A. O., Borghesani, V., Pashkov, A., Marinazzo, D., Nicholas, J., Salatiello, A., Sucholutsky, I., Minervini, P., Razavi, S., Rocca, R., Yusifov, E., Okalova, T., Gu, N., Ferianc, M., Khona, M., Patil, K. R., Lee, P.-S., Mata, R., Myers, N. E., Bizley, J. K., Musslick, S., Bilgin, I. P., Niso, G., Ales, J. M., Gaebler, M., Murty, N. A. R., Loued-Khenissi, L., Behler, A., Hall, C. M., Dafflon, J., Bao, S. D., & Love, B. C. (2024). Large language models surpass human experts in predicting neuroscience results. arXiv. 2403.03230v2 [q-bio.NC]. https://doi.org/10.48550/arXiv.2403.03230

➔ *An example of how LLMs can perform tasks better than humans.*

Meta (2020). Retrieval Augmented Generation: Streamlining the creation of intelligent natural language processing models. https://ai.meta.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/

➔ *A Meta article that more succinctly explains how RAG works and how it is implemented in the LLM pipeline.*

Monigatti, L. (2023, November 14). Retrieval-Augmented Generation (RAG): From Theory to LangChain Implementation. Towards Data Science. https://towardsdatascience.com/retrieval-augmented-generation-rag-from-theory-to-langchain-implementation-4e9bd5f6a4f2

➔ *This article provides a wonderful illustration of the RAG process that has been featured in this paper.*

Provost, F., & Fawcett, T. (2013). Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. O'Reilly Media.

➔ *A useful textbook that explains the history and basics of data science*

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2023). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv. 2201.11903(6). https://doi.org/10.48550/arXiv.2201.11903

➔ *This paper introduced Chain of Thought prompting and is the original source of this now widely used technique.*

Wei, J., Yang, C., Song, X., Lu, Y., Hu, N., Huang, J., Tran, D., Peng, D., Liu, R., Huang, D., Du, C., & Le, Q. V. (2024). Long-form factuality in large language models. arXiv:2403.18802 [cs.CL]. https://doi.org/10.48550/arXiv.2403.18802

➔ *Another example of how LLMs perform better than humans in certain tasks.*

Welivita, K. A., & Pu, P. (2024). Is ChatGPT More Empathetic than Humans? arXiv:2403.05572 [cs.HC]. https://doi.org/10.48550/arXiv.2403.05572

➔ *Another example of how LLMs perform better than humans in certain tasks.*

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. 2302.11382(1). https://doi.org/10.48550/arXiv.2302.11382

➔ *An article that offers a very thorough prompting pattern guide for working in various domains using ChatGPT. White et al explain the need to incorporate intent and motivation in prompts. This will be useful when choosing the necessary wording during prompt development.*

Zhang, H., Wu, C., Xie, J., Lyu, Y., Cai, J., & Carroll, J. M. (2023). Redefining Qualitative Analysis in the AI Era: Utilizing ChatGPT for Efficient Thematic Analysis. arXiv. 2309.10771 [cs.HC]. https://doi.org/10.48550/arXiv.2309.10771

➔ *This article addresses the same topic we discuss in this paper: using LLMs in thematic analysis. Zhang et al. identify risks involved with using LLMs in thematic analysis applications and attitudes by researchers when attempting to incorporate LLMs into their workflows.*

10. Appendix

Table 1

Comparison of Qualitative Analysis Software

Thematic Analysis AI Software Comparison				
	MAXQDA	**Atlas.ti**	**NVivo**	**CoLoop**
Powered by:	OpenAI	OpenAI	Self (Lumivero)	Genei
Able to Summarise	• documents (text, transcript, etc) • coded segments of a topic • selected text	• documents (text, transcript, etc) • key concepts or descriptive themes in data	• documents (text, transcript, etc) • key concepts or descriptive themes in data	• interviews, activities, participants or tasks • demographics • other individual questions in the AI chat interface
Coding	• can suggest new codes and subcodes	• can identify codes in documents • can suggest new codes based on research scope and intentions • codes can be fine-tuned to necessary granularity	• can identify codes in documents • code a portion of the data and Nvivo uses code patterns to do the rest • assigns codes for sentiment analysis	• allows for tagging within the data • an analysis grid contains prompts to help user identify themes • coding framework not explicitly provided
Positives	• Can explain selected words or text passages • Many ways to customise the type and length of summarization	• offers analytical tools to query data such as code distribution which can then be visualised • AI chat available to converse naturally with data	• greater privacy/security by using internal tools (not ChatGPT) • no hallucinations • data visualisation capabilities	• Can add intention to queries with project description • possibilities are endless for insights with chat interface • quotes provided by chat to support answers • use composed pipelines of fine-tuned models to limit hallucinations and always refer back to the primary data
Negatives	• may get slightly varying summaries for the same material • social biases could affect results	• automatically generated codes may differ for the same material • social biases could affect results • data must be formatted into clearly defined paragraphs to work • AI Coding skips very short paragraphs. • Existing codes, quotations, and formatting have no bearing on AI results	• lacks natural language capabilities in AI	• need experimentation with chat prompts to achieve desired results • if chat is not reset enough, the quotes are not relevant to the prompt topic
When AI assistance takes place:	• after human coding takes place	• before human coding and then human correction takes place after	• before or after human coding takes place	• codes are identified organically through conversation with AI
Context Window	60,000	6000 to 12,000 words (depending on language and text formatting) for trial	N/A	Unknown
Feedback	"summaries generated do not always capture the nuance contained within the data segments collected at the code…keeps a good balance between the role of the AI and the human interpreter" [1]	"currently implemented, is for such purposes – namely to quickly explore text at a high level as a means of becoming familiar, potentially useful as a precursor to (but not as yet a replacement of) human coding"[2]	"efficiently manage and organise large volumes of qualitative data…cost of acquiring NVivo licences and the need for additional training might pose financial and time constraints, particularly for individual researchers"[3]	"Although CoLoop is pretty good at generating summaries based on segments, what I look forward to is developments that allow the material to be tagged such that patterns and relationships beyond descriptive summaries can be accessed and interrogated in further detail"[4]

[2] https://www.qdaservices.co.uk/post/ai-assist-beta-from-maxqda-what-s-it-good-for
[3] https://www.qdaservices.co.uk/post/ai-open-coding-beta-from-atlas-ti-what-s-it-good-for
[4] https://www.researchgate.net/publication/372912868_The_Impact_of_NVivo_in_Qualitative_Research_Perspectives_from_Graduate_Students
[5] https://www.qdaservices.co.uk/post/the-ai-copilot-from-coloop-what-s-it-good-for