A New Method Supporting Qualitative Data Analysis Through Prompt Generation for Inductive Coding

Fengxiang Zhao[1] Fan Yu[2] Yi Shang[1]
[1]*Department of Electrical Engineering and Computer Science, University of Missouri-Columbia, USA*
[2]*School of Information Science and Learning Technogies, University of Missouri-Columbia, USA*
{fzfmx, fybxd, shangy}@missouri.edu

***Abstract*—Recent advances in Large Language Models (LLMs) have revolutionized numerous fields, including Qualitative Data Analysis (QDA). This paper introduces a novel method, ARGUMENT2CODE (A2C), designed to leverage the capabilities of LLMs for enhancing the QDA process, particularly focusing on the inductive coding aspect. A2C sets itself apart from conventional automated coding tools by initiating a two-stage fine-tuned LLM process adept at navigating the complex landscape of qualitative data. This innovative method starts with the identification of coding cues hidden within the textual data, which are then refined into targeted prompts. These prompts are then used for guiding the inductive coding process, facilitating the generation of a rich, actionable codebook that offers expansive coverage of analytical perspectives. Our experimentation reveals that A2C not only successfully generates a pertinent and insightful codebook consistently but also significantly outperforms all other existing methods.**

***Index Terms*—Qualitative Data Analysis, Inductive Coding, Large Language Models (LLMs), Textual Data Analysis, Thematic Analysis.**

I. INTRODUCTION

Qualitative coding serves as a cornerstone in qualitative research, enabling the distillation of complex, non-numeric data into discernible patterns and categories. This method is critical for extracting insights from diverse textual corpora, including social media content, open-ended survey responses, and interview transcripts, which often encapsulate nuanced human experiences and perceptions [1]. Traditionally, this labor-intensive process demands meticulous effort from researchers to develop comprehensive codebooks and achieve inter-rater reliability. The advent of large language models (LLMs) presents a novel approach to this challenging task. Recent studies indicate that LLMs can significantly enhance the efficiency and accuracy of qualitative coding by automating initial coding phases, thus allowing researchers to focus on higher-order analysis and theory development. This integration of advanced computational tools within qualitative research paradigms promises to not only expedite the analytical process but also enhance the depth of theoretical insights derived from vast datasets [2] [3].

The primary motivation for this research is rooted in the limitations of traditional coding methods in qualitative research, which, despite some automation, still require substantial manual effort. The advent of LLMs presents a promising solution to these challenges by potentially automating the coding process more effectively. This study aims to leverage LLMs to fully automate inductive coding, aiming to retain the depth of analysis provided by manual methods while significantly enhancing efficiency and scalability. By integrating LLMs, the research seeks to minimize human intervention, streamline the coding process, and enable qualitative researchers to manage larger datasets more effectively. The proposed method, ARGUMENT2CODE (A2C), leveraging the power of LLMs to captures the complexity and richness of qualitative data. Our focus on prompt generation for inductive coding is driven by the belief that the right prompts can serve as powerful tools for guiding the analytical process, enabling researchers to uncover deeper insights and foster a more engaging interaction with their data.

The major contributions of this paper are summarized below:

- We introduce a novel method ARGUMENT2CODE to generate a comprehensive codebook from textual arguments, leveraging the deep contextual understanding offered by modern LLMs, such as ChatGPT. This method represents a significant step forward in automating and enhancing the inductive coding process in QDA.
- We design a new metric for evaluating the coverage of the generated codebook. This metric enables a precise and meaningful comparison of our approach to human-generated codebooks and other automated methods, highlighting the depth and comprehensiveness of our method.
- We conduct comprehensive experiments to compare A2C with various existing methods, demonstrating its superior performance in generating relevant and insightful codes.

II. RELATED WORK

The emergence of deep learning has revolutionized various sectors by providing groundbreaking solutions and reshaping our approach to diverse challenges. It has notably transformed industries such as healthcare [4], agriculture [5], and finance [6], among others. Similarly, the development of large language models (LLMs) has dramatically improved qualitative

2835-5776/24/$31.00 ©2024 IEEE
DOI 10.1109/IRI62200.2024.00043

data analysis (QDA), helping researchers process and analyze data in ways that were previously unattainable.

A. LLMs in Qualitative Data Analysis

There are two distinct approaches to qualitative coding: deductive and inductive [7]. Deductive coding involves using pre-established codes to either confirm or refute a hypothesis during the analysis process. Conversely, inductive coding entails creating codes spontaneously as documents are reviewed, allowing for the identification of new phenomena within the data.

1) Deductive Coding: Many AI-based tools has been built to assist QDA [8] [9]. In recent research, LLMs such as GPT-3 have been effectively employed to assist with deductive coding tasks in qualitative research. For instance, Xiao et al. [10] demonstrated that combining LLMs with expert-drafted codebooks can achieve substantial agreement with expert-coded results in various datasets. [11] highlights the potential of ChatGPT has great potential to serve as an augmentative tool rather than a replacement for the intricate analytical tasks performed by humans.

2) Inductive Coding: Efforts have been made to enhance the inductive coding process through various technological advancements. Research has investigated the integration of semi-automation in qualitative research, developing prototypes that matched human coding reliability using natural language processing [12]. Other studies have explored the temporal and perspective roles in human-AI collaboration, emphasizing tool placement in enhancing qualitative analysis efficiency [13]. The use of machine learning to support qualitative coding has shown promising results in addressing data ambiguity, demonstrating how AI can improve both accuracy and efficiency in social science research [14]. The introduction of rule-based learning and supervised learning techniques has been a milestone in automating the inductive coding process [15] [16]. A method combining human-defined coding rules with LLMs to suggest codes for qualitative data analysis has been proposed, speeding up the coding process while enhancing the accuracy and adaptability of code suggestions through user interactions [17]. Another novel approach uses LLMs for qualitative data analysis, thereby improving the analysis process by efficiently identifying relevant and significant keypoints [18]. Lastly, an experiment with GPT-3.5-Turbo to emulate aspects of an inductive thematic analysis has been discussed, examining the potential for LLMs to partially infer main themes from qualitative data and reflecting on both the capabilities and limitations of LLMs within a qualitative research context [19].

B. Instruction Tuning for LLMs

Instruction tuning involves additional training of Large Language Models (LLMs) using a dataset composed of $(INSTRUCTION, OUTPUT)$ pairs in a supervised manner. This process helps align the LLMs' original next-word prediction goal with the users' goal of ensuring the models follow human instructions effectively [20]. Significant advancements highlight the improvements in zero-shot capabilities through instruction tuning using machine-generated data, which significantly boosts LLMs performance on tasks like Alpaca (7B) [21] or tuning with GPT-4-generated data enhances zero-shot performance on novel tasks, underscoring the potential of machine-generated instructions in refining LLMs [22]. The diversity and quality of instructions are crucial, as demonstrated by efforts to distill instruction-tuned knowledge into smaller models while ensuring a broad coverage of topics to maintain effectiveness [23]. Behavioral insights from instruction tuning reveal improved instruction recognition and better alignment of LLMs' knowledge with user-intended tasks, enhancing response quality [24].

III. Problem Definition

The aim of inductive coding is to construct a comprehensive codebook from text data. This codebook is essential for analyzing qualitative data as it enables the identification of themes, patterns, and relationships within the data.

Formally, given a set of textual arguments $A = a_1, a_2, ..., a_n$, where each argument a_i is a text entry, our goal is to generate a codebook $C = c_1, c_2, ..., c_k$, where each code c_j is a category derived from the content of the arguments in A, with k being a predefined number representing the desired number of unique codes.

The coverage metric is used to quantify the extent to which the generated codes capture the thematic or conceptual breadth of reference codes. Coverage is defined by evaluating the similarity of each generated code to its best-matching reference code, incorporating adjustments for length discrepancies between the generated and reference codes. A high coverage score indicates effective representation of key aspects of the reference sets, suggesting a comprehensive understanding and replication of essential thematic elements. Conversely, a lower score points to an inadequate representation.

Specifically, consider a collection of generated codes $\hat{R} = \hat{r}_1, \hat{r}_2, ..., \hat{r}_m$ and a set of reference codes $R = R_1, R_2, ..., R_m$, our objective is to maximize the representational breadth of $\hat{R}$ relative to R. This maximization is conducted using BertScore to quantify the semantic correspondence between two pieces of text while adjusting for the length similarity between the generated and reference codes to ensure the fairness of comparison.

As shown in Algorithm 1, the matching process employs the Hungarian Algorithm [25] to find the best correspondence between $\hat{R}$ and R, maximizing overall similarity. $\text{sim}(a, b)$ is the BertScore similarity score between code a and code b.

After matching, coverage is computed to measure how comprehensively the generated codes represent the thematic elements found in the reference set while adjusting for length to avoid penalizing or unfairly rewarding discrepancies in code length.

The formula for calculating coverage, adjusted for length difference, is given by:

$$\text{Coverage}(\hat{R}, R) = \frac{1}{m} \sum_{i=1}^{m} \text{sim}(\hat{r}_i, r_i^*) \times L(\hat{r}_i, r_i^*)$$

where

Algorithm 1 Simplified Optimal Matching of Codes

Inputs: Generated codes $\hat{R}$, Reference codes R
Outputs: Matched pairs with scores
Initialize empty cost matrix C
for each generated code $\hat{r}_i$ in $\hat{R}$ **do**
  for each reference code r_j in R **do**
    $C[i][j] \leftarrow -\text{sim}(\hat{r}_i, r_j)$
  end for
end for
Apply Hungarian Algorithm on C to find matches
return Matches and their similarity scores

- $\hat{r}_i$ is the i-th generated code.
- r_i^* represents the reference code that has been identified as the best match for $\hat{r}_i$ in terms of both semantic similarity.
- $\text{sim}(a, b)$ calculates the BertScore similarity score between code a and code b.
- $L(a, b)$ represents the length penalty between code a and code b, optimizing for codes that closely match the length of their references.

The length penalty function is defined as:

$$L(a, b) = \exp\left(\left|\frac{\text{len}(a) - \text{len}(b)}{\text{len}(b)}\right|\right)$$

where $\text{len}(a)$ and $\text{len}(b)$ are the number of tokens (or characters, depending on the context) in code a and b, respectively. This function approaches 1 (the lower bound) when the lengths of a and b are close, implying minimal penalty for length discrepancies, thus promoting length similarity between the generated code and its reference.

Incorporating the length penalty term ensures that the coverage metric rewards comprehensive and accurately represented codes that are also contextually concise, mirroring the optimal length of the reference codes. This approach mitigates the potential biases associated with length disparities, promoting a more balanced and fair assessment of code generation quality.

IV. Argument2Keypoint for Inductive Coding

In this section, we present the new Argument2Code (A2C) method designed to facilitate inductive coding in QDA. A2C is distinctive for its structured approach in identifying and generating codebook from complex textual data, thereby enhancing the breadth and depth of thematic analysis.

A. Overview

A2C consists of a two-stage process, pivoting around the generation and utilization of Analytical Guide Prompts for inductive coding:

1) **Generation of Analytical Guide Prompts:** Initially, a Large Language Model (LLM) , denoted as

$$M_a : \{a^1, ..., a^k\} \rightarrow p$$

is fine-tuned to generate an analytical guide prompt p from arguments $a^1, ..., a^k$. Generated prompt is designed to encapsulate potential themes and guide and serve as a scaffold for deeper thematic exploration.

2) **Generation of Codes:** Following the identification of a suitable analytical guide prompt, another LLM , denoted as

$$M_p : (\{a^1, ..., a^k\}, p) \rightarrow \{c^1, ..., c^m\}$$

is fine-tuned to generate thematic codes $\{c^1, ..., c^m\}$ for the set of arguments $\{a^1, ..., a^k\}$ in relation to an analytic guide prompt p. This ensures that the thematic analysis is anchored to the initially identified prompts, yielding a thematic output that is both tailored and contextually informed for each argument.

To code a set of arguments $\{a'^1, ..., a'^k\}$, M_a is initially utilized to identify relevant analytical guide prompts. These prompts are then employed to query M_p, which generates codes for arguments with the guide from analytical guide prompt. This dual-stage methodology not only guarantees extensive theme coverage but also circumvents the common pitfalls of generic and broad code generation found in single-stage coding approaches.

To facilitate effective fine-tuning and inference with the necessary depth of contextual information, we implement advanced techniques such as LoRA [26] that focuses on the efficiency of updates by modifying low-rank matrices exclusively. For any given pre-trained weight matrix $W \in \mathbb{R}^{d\times k}$ of a LLM, such as a transformer neural network model, LoRA recommends a decomposition $W = W + BA$, with $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, and $r << \min(d, k)$. Throughout training of a LLM, W is held constant, and adjustments are made to A and B. In a Transformer's context, LoRA selectively refines attention weights W_q, W_k, W_v, W_o while maintaining the rest of the parameters static.

B. Prompt Generation Pipeline for Inductive Code Development

In the first stage of A2C, we propose a new prompt generation pipeline specifically designed to generate a single, comprehensive analytical guide prompt from a dataset of arguments $A = \{a_1, a_2, ..., a_n\}$, as illustrated in Fig 1 . This prompt is instrumental in facilitating the subsequent generation of codes, enhancing the depth and relevance of analysis across various thematic dimensions.

The primary goal of the pipeline is to craft an analytical guide prompt that captures multiple dimensions of a complex issue comprehensively. This guide prompt plays a crucial role in structuring the creation of detailed and focused codes, thereby ensuring a thorough examination of the topic under study.

An analytical guide prompt is a structured set of instructions aimed at thoroughly exploring a specific topic. It typically encompasses several components that encourage the analysis of different dimensions, such as ethical, legal, societal, and medical aspects. By addressing these categories, the prompt fosters a balanced and profound discussion, making sure that both supporting and opposing viewpoints are considered.

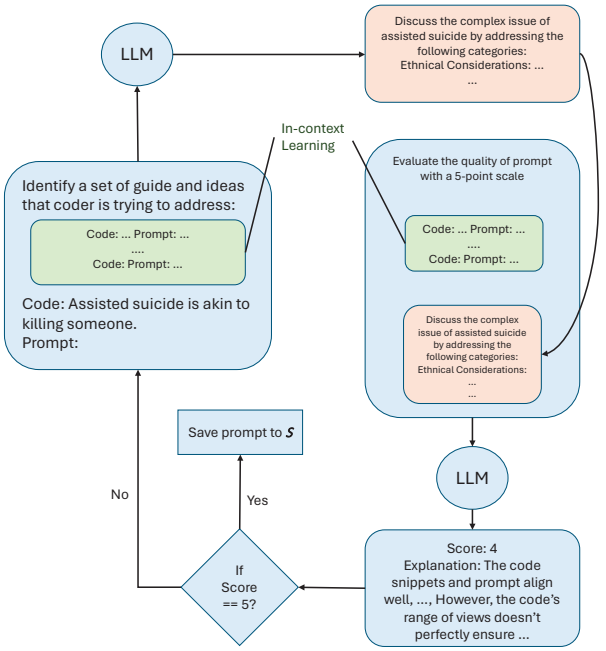

Fig. 1. The new prompt generation pipeline in Stage 1 of A2C.

The main steps of the pipeline are as follows.

1) **Generation Step:** Initiated by populating a set S of human-annotated examples that act as initial contextual examples for generation. Employing the Llama-2-13B-Chat [27] model, we generate prompts for a randomly selected subset of 10 codes. These prompts undergo manual refinement to ensure their relevance and insightful nature. The in-context learning (ICL) [28] strategy deployed here dynamically leverages examples from S, maintaining the diversity and relevance of generated prompts while conforming to context length constraints.
2) **Evaluation Step:** Using chain-of-thought prompting [29], we use Llama-2-13B-Chat model to evaluate each generated prompt on a 5-point scale. To standardize evaluations, five in-context examples for each grade from 1 to 5 are utilized.
3) **Regeneration Strategy:** When a prompt scores below 5, it enters a phase of regeneration. This cycle avoids repetition by utilizing a dynamically selected set of contextual examples. A prompt may undergo up to five regeneration iterations per code.
4) **Output:** The refined set S, now encompassing code-prompt pairs (x_i^n, c_i^n) with prompt x_i^n of score 5 for code c_i^n, is outputted for subsequent analysis.

Through this pipeline, the creation of high-quality, pertinent prompts is assured.

TABLE I
EXAMPLES OF ARGKP-2021 DATASET OF ARGUMENTS AND KEY POINTS ON THE TOPIC OF ASSISTED SUICIDE.

Topic	Argument	Key Point
Assisted suicide should be a criminal offence	A cure or treatment may be discovered shortly after having ended someone's life unnecessarily.	Assisted suicide allows people to solicit someone to die to their own benefit
Assisted suicide should be a criminal offence	A cure or treatment may be discovered shortly after having ended someone's life unnecessarily.	Assisted suicide is akin to killing someone
Assisted suicide should be a criminal offence	A patient should be able to decide when they have had enough "care".	Assisted suicide reduces suffering

V. EXPERIMENTS AND RESULTS

A. Dataset

We employed the ArgKP-2021 dataset [30], which is tailored to advance research in key point analysis and argument mining. We randomly select 70% of the corpus for training and 30% for testing.

The ArgKP-2021 dataset is organized around a set of 29 debatable topics. Each topic serves as a thematic focus for the arguments and is associated with approximately 300 individual arguments and about 10 key points, though the number of key points per topic can range from 4 to 14. The dataset's structure is defined by triplets formatted as topic, argument, and key point:

- **Topic:** Acts as the central theme around which arguments are developed, representing areas of significant public interest or debate.
- **Argument:** Presents an individual's perspective or stance on the topic, offering a diverse array of viewpoints which enriches the dataset's complexity and depth.
- **Key Point:** Provides a concise, expertly crafted summary that encapsulates the core essence of one or more arguments, facilitating clearer and more streamlined analysis of varied perspectives.

Table I shows examples of ArgKP-2021 dataset of arguments and key points on the topic of assisted suicide.

B. Experimental Results

We fine-tuned Llama-2-13B-Chat [27] as the instruction-following model for prompt generation and Llama-2-7B-Chat as stage 2 model in A2C using LLaMa-Factory [31].

1) Keypoint Generation: To evaluate the effectiveness of our generated keypoints in comparison to the reference keypoints, we utilize three commonly used performance metrics: BLEU [32], ROUGE [33], and BertScore [34]. BLEU and ROUGE are used to assess the n-gram overlap, which quantifies the lexical similarity between the generated and reference prompts. BertScore, on the other hand, evaluates the semantic similarity by analyzing how closely the embeddings of the

generated prompts align with those of the reference prompts. We compare A2C with the following methods:

- **Luhn's Algorithm [35]:** An early approach to automatic summarization, Luhn's Algorithm focuses on the significance of high-frequency words within a set context window. This heuristic method serves as a historical benchmark.
- **Direct Inference (DI):** This method directly uses Llama-2-7B-Chat without any fine-tuning or the use of an analytical guide prompt, acts as an ablation setup to evaluate the effects of fine-tuning in A2C.
- **Fine-Tuned Direct Inference (FT-DI):** This fine-tuned Llama-2-7B-Chat model serves as a single stage model to generate codes from arguments. This method mirrors commonly employed pipelines in previous papers and serves as a baseline representing the state of the art in machine learning practices for qualitative data analysis.

Table II shows performance comparison of ARGUMENT2CODE (A2C) with other methods. A2C achieved improvement in the extraction and formulation of qualitative codes across all metrics compared to Luhn's Algorithm, Direct Inference (DI), and Fine-Tuned Direct Inference (FT-DI). Specifically, A2C scores the highest in BLEU, ROUGE (1, 2, and L), and BertScore metrics, which signifies its effectiveness in generating more semantically relevant and lexically aligned codes with respect to the reference codes. The performance uplift can be attributed to the two-stage process of A2C, which fine-tunes the language model to generate analytical guide prompts that direct the code generation process, ensuring higher quality and more contextually appropriate codes.

2) Coverage: Table II shows that A2C outperforms the other methods with a coverage score of 64.53. This higher coverage score reflects the method's ability to comprehensively represent the thematic elements found in the reference set. A2C's strategy of generating analytical guide prompts is crucial for this performance, as it results in a broader exploration of the thematic space within the dataset. By guiding the coding process through these prompts, A2C ensures a balanced thematic representation. Such a comprehensive coverage is pivotal for qualitative analyses, where capturing a wide range of themes and subtleties in the data is crucial for depth and richness of insight.

3) Ablation Study: The ablation study aims to isolate and evaluate the impact of key components of the A2C method. By selectively deactivating certain features such as the generation of analytical guide prompts (i.e., comparing A2C with the Direct Inference and Fine-Tuned Direct Inference baselines), we assessed their individual contributions to the overall performance. The study reveals that the removal of the prompt generation and fine-tuning process results in noticeable decrements in performance across all evaluation metrics. Particularly, the comparison between A2C and FT-DI illustrates the added value of incorporating guide prompts into the code generation process, not just the effect of fine-tuning alone. This aligns with our hypothesis that the prompt-based approach contributes to generating more nuanced and context-aware codes. The findings from the ablation study underscore the importance of both stages in the A2C process, validating the efficacy of employing analytical guide prompts for enhanced thematic exploration in inductive code generation.

TABLE II
RESULTS OF FOUR DIFFERENT METHODS ON 6 PERFORMANCE METRICS (DI: DIRECT INFERENCE, FT-DI: FINE-TUNED DIRECT INFERENCE, A2C: ARGUMENT2CODE).

Method	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	BertScore	Coverage
Luhn's Algorithm	2.23	7.94	8.86	17.52	85.71	15.62
DI	2.30	4.07	7.04	17.94	85.75	59.75
FT-DI	2.23	31.11	10.08	19.58	86.45	60.75
A2C	**4.13**	**37.98**	**11.73**	**20.72**	**86.89**	**64.53**

These preliminary results show that A2C offers a promosing tool for qualitative data analysis. The method's capability to produce relevant, comprehensive, and nuanced codes holds considerable promise for supporting qualitative researchers in navigating large textual datasets. By streamlining the inductive coding process without compromising on analytical depth, A2C stands as a advancement in the integration of large language models within qualitative research methodologies.

VI. CONCLUSION AND DISCUSSION

This paper introduced ARGUMENT2CODE (A2C), a novel method aimed at enhancing the capabilities of qualitative data analysis (QDA) through the use of Large Language Models (LLMs) for generating inductive codes. By incorporating a two-stage approach that combines the generation of analytical guide prompts with the subsequent generation of thematic codes, A2C addresses the existing gap in automated tools for qualitative analysis – namely, the need for nuanced understanding and contextual appropriateness in the coding process.

Our experimental results, validated across standard metrics such as BLEU, ROUGE, BertScore, and a newly introduced coverage metric, shows that A2C outperformed traditional methods and direct applications of LLMs in generating relevant and comprehensive codes from textual data. The superiority of A2C in terms of performance, coverage, and specificity highlights its potential to serve as a valuable asset for researchers engaged in QDA.

This study has advanced the use of LLMs for generating inductive codes automatically, but it faces significant limitations. The fully automated ARGUMENT2CODE (A2C) method enhances efficiency for large datasets, yet it omits the crucial human review, potentially missing the nuanced multi-dimensionality and depth inherent in qualitative data due to its need for subjective judgment and interpretation [36]. Additionally, the absence of a dedicated public dataset for inductive coding necessitated reliance on approximations for validating the A2C method's effectiveness, which may not accurately represent its performance in real-world qualitative research settings. Further research is required to validate the A2C framework with richer, more specialized datasets to confirm its efficacy and applicability.

In conclusion, ARGUMENT2CODE represents a step forward in the integration of advanced AI technologies within the domain of qualitative research. By providing a method that

enhances both the efficiency and depth of qualitative data analysis, A2C paves the way for more nuanced, comprehensive, and insightful exploration of qualitative datasets. As AI technologies continue to progress, the potential for such methods to transform qualitative research methodologies – making them more accessible, thorough, and insightful – is immense. Future research will undoubtedly expand on these foundations, exploring new ways to synergize human intuition and AI's computational power in the pursuit of deeper understanding within the qualitative research paradigm.

REFERENCES

[1] P. Burnard, "A method of analysing interview transcripts in qualitative research," *Nurse Education Today*, vol. 11, no. 6, pp. 461–466, 1991.

[2] V. Elliott, "Thinking about the Coding Process in Qualitative Data Analysis," *The Qualitative Report*, Nov. 2018.

[3] A. Castleberry and A. Nolen, "Thematic analysis of qualitative research data: Is it as easy as it sounds?" *Currents in Pharmacy Teaching and Learning*, vol. 10, no. 6, pp. 807–815, Jun. 2018.

[4] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 24–29, Jan. 2019.

[5] F. Tian, C. J. Ransom, J. Zhou, B. Wilson, and K. A. Sudduth, "Assessing the impact of soil and field conditions on cotton crop emergence using uav-based imagery," *Computers and Electronics in Agriculture*, vol. 218, p. 108738, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169924001297

[6] J. B. Heaton, N. G. Polson, and J. Witte, "Deep learning for finance: Deep portfolios," *ERN: Other Econometrics: Computer Progr(Topic)*, 2016.

[7] S. Haug, T. Rietz, and A. Maedche, "Accelerating deductive coding of qualitative data: An experimental study on the applicability of crowdsourcing," in *Proceedings of Mensch Und Computer 2021*, ser. MuC '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 432–443. [Online]. Available: https://doi.org/10.1145/3473856.3473873

[8] T. Rietz and A. Maedche, "Cody: An ai-based system to semi-automate coding for qualitative research," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, ser. CHI '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3411764.3445591

[9] P. Paredes, C. Cheshire, A. S. Rufino Ferreira, C. Schillaci, G. Yoo, D. Xing, P. Karashchuk, and J. Canny, "Inquire: Large-scale early insight discovery for qualitative research," 02 2017.

[10] Z. Xiao, X. Yuan, Q. V. Liao, R. Abdelghani, and P.-Y. Oudeyer, "Supporting Qualitative Analysis with Large Language Models: Combining Codebook with GPT-3 for Deductive Coding," in *28th International Conference on Intelligent User Interfaces*. Sydney NSW Australia: ACM, Mar. 2023, pp. 75–78.

[11] M. S. Jalali and A. Akhavan, "Integrating ai language models in qualitative research: Replicating interview data analysis with chatgpt," Available at SSRN: https://ssrn.com/abstract=4714998 or http://dx.doi.org/10.2139/ssrn.4714998, Feb 2024, accessed: date-of-access.

[12] M. Marathe and K. Toyama, "Semi-automated coding for qualitative research: A user-centered inquiry and initial prototypes," *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018.

[13] J. L. Feuston and J. R. Brubaker, "Putting tools in their place: The role of time and perspective in human-ai collaboration for qualitative analysis," *Proc. ACM Hum.-Comput. Interact.*, vol. 5, no. CSCW2, oct 2021. [Online]. Available: https://doi.org/10.1145/3479856

[14] N.-C. Chen, M. Drouhard, R. Kocielnik, J. Suh, and C. Aragon, "Using machine learning to support qualitative coding in social science: Shifting the focus to ambiguity," *ACM Transactions on Interactive Intelligent Systems*, vol. 8, pp. 1–20, 06 2018.

[15] K. Crowston, X. Liu, and E. Allen, "Machine learning and rule-based automated coding of qualitative data," pp. 1–2, 2010.

[16] M. Scharkow, "Thematic content analysis using supervised machine learning: An empirical evaluation using German online news," *Quality & Quantity*, vol. 47, no. 2, pp. 761–773, Feb. 2013.

[17] C. Spinoso-Di Piano, S. Rahimi, and J. Cheung, "Qualitative Code Suggestion: A Human-Centric Approach to Qualitative Coding," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 14 887–14 909.

[18] F. Zhao *et al.*, "A new method using llms for keypoints generation in qualitative data analysis," in *2023 IEEE Conference on Artificial Intelligence (CAI)*, 2023, pp. 333–334.

[19] S. Paoli, "Can large language models emulate an inductive thematic analysis of semi-structured interviews? an exploration and provocation on the limits of the approach and the model," *ArXiv*, vol. abs/2305.13014, 2023.

[20] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, and G. Wang, "Instruction tuning for large language models: A survey," 2024.

[21] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Alpaca: A strong, replicable instruction-following model," *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, vol. 3, no. 6, p. 7, 2023.

[22] B. Peng, C. Li, P. He, M. Galley, and J. Gao, "Instruction tuning with gpt-4," *ArXiv*, vol. abs/2304.03277, 2023.

[23] M. Wu, A. Waheed, C. Zhang, M. Abdul-Mageed, and A. F. Aji, "Lamini-lm: A diverse herd of distilled models from large-scale instructions," *ArXiv*, vol. abs/2304.14402, 2023.

[24] X. Wu, W. Yao, J. Chen, X. Pan, X. Wang, N. Liu, and D. Yu, "From language modeling to instruction following: Understanding the behavior shift in llms after instruction tuning," *ArXiv*, vol. abs/2310.00492, 2023.

[25] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109

[26] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[27] H. Touvron *et al.*, "Llama 2: Open foundation and fine-tuned chat models," Jul. 2023.

[28] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui, "A survey on in-context learning," 2023.

[29] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023.

[30] R. Bar-Haim, L. Eden, R. Friedman, Y. Kantor, D. Lahav, and N. Slonim, "From arguments to key points: Towards automatic argument summarization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 4029–4039. [Online]. Available: https://aclanthology.org/2020.acl-main.371

[31] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, Z. Luo, and Y. Ma, "Llamafactory: Unified efficient fine-tuning of 100+ language models," *arXiv preprint arXiv:2403.13372*, 2024. [Online]. Available: http://arxiv.org/abs/2403.13372

[32] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Annual Meeting of the Association for Computational Linguistics*, 2002. [Online]. Available: https://api.semanticscholar.org/CorpusID:11080756

[33] C.-Y. Lin, "Automatic evaluation of summaries using n-gram co-occurrence statistics," 01 2003, pp. 71–78.

[34] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with BERT," *CoRR*, vol. abs/1904.09675, 2019. [Online]. Available: http://arxiv.org/abs/1904.09675

[35] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.

[36] J. Jiang, K. Wade, C. Fiesler, and J. Brubaker, "Supporting serendipity: Opportunities and challenges for human-ai collaboration in qualitative analysis," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, pp. 1–23, 04 2021.