

Leveraging LLMs to Refine Codebooks for Analyzing Computing Students' Reflections on Coursework

Amruth N. Kumar^{1,*†}, Priscilla Jimenez-Pazmino^{2,†}, Tammy VanDeGrift^{3,†} and Bonnie K MacKellar^{4,†}

¹Ramapo College of New Jersey, Mahwah, NJ 07430, USA

²University of St. Thomas, St. Paul, MN 55105, USA

³University of Portland, Portland, OR 97203, USA

⁴St. John's University, New York, NY 11439, USA

Abstract

Can Large Language Models (LLM) help refine codebooks used for deductive coding of text data? We propose a Leave-one-out approach to refine codebooks in which all the themes in the codebook are left out one at a time and an LLM is asked to suggest additional themes after using the modified codebook to deductively code data. The suggested additional themes are then analyzed to aid in the refinement of the codebook. This approach is systematic, and hence generalizable and replicable. We conducted a pilot study using data collected from computing students on how they were persistent when completing course assignments. Expert researchers had created a codebook of seven themes for the data that was used in the evaluation of the approach. We found that in three cases, LLM (Claude Sonnet 4.6) suggested re-inclusion of the omitted theme in the very run in which the theme had been omitted. Further, it suggested splitting two composite themes in the original codebook into three sub-themes each. Finally, it recommended including two additional themes influenced by the specifics of the data set. We propose two measures: statement-stability score to identify candidates for exemplars of a theme, and theme-stability score that quantifies the cohesion of a theme. This work-in-progress pilot suggests that LLMs can indeed help researchers iteratively refine codebooks using the Leave-one-out approach.

Keywords

Large Language Model, LLM, Qualitative analysis, Codebook refinement, Persistence

1. Introduction

Qualitative analysis is used in educational research to analyze text data. It may be inductive in nature, involving the iterative identification of themes in the data [1]. It may also be deductive, in which existing themes, models, or theories are used to categorize the data. A major issue for qualitative coding approaches is that they are time- and labor-intensive [2]. Because of these factors, qualitative coding does not easily scale to large data sets. Therefore, researchers have been investigating whether and how to use LLMs, themselves trained on massive amounts of text, to assist in qualitative coding.

A codebook is a coding framework typically used to deductively code text data. It contains a set of themes, along with their definitions, and optionally, exemplars and decision rules. Even after a codebook is created, it may need to be periodically reviewed and refined, either because of accumulation of additional data or because of the need to apply it to a data set collected in a new context. For example, with computer science educational data, the student skillset has changed over time: programming is now a standard part of K12 curriculum [3]. The tools available for programming have changed dramatically, due to the availability of better IDEs, more detailed online sources (stackoverflow), and now, LLMs (Claude Code, CoPilot, etc.) [4]. Therefore, computer science data sets change over time and consequently, codebooks to analyze them must be refined from time to time.

CSEDM'26: 10th Educational Data Mining in Computer Science Education (CSEDM) Workshop, June 28, 2026, Seoul, South Korea

*Corresponding author.

†These authors contributed equally.

✉ amruth@ramapo.edu (A. N. Kumar); priscilla.jimenez@stthomas.edu (P. Jimenez-Pazmino); vandegri@up.edu (T. VanDeGrift); mackellb@stjohns.edu (B. K. MacKellar)

ORCID 0000-0002-1951-3995 (A. N. Kumar); 0000-0002-3605-3011 (P. Jimenez-Pazmino); 0000-0002-8054-7588 (T. VanDeGrift); 0000-0003-2269-5029 (B. K. MacKellar)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

An advantage of involving LLMs to refine codebooks is that LLMs may bring a broader perspective to the task because they have been trained on massive amounts of data [5]. Another is that they may save time and effort that would otherwise be needed for an entirely manual refinement of codebook. Can we leverage LLMs to help computer science education researchers refine existing codebooks? One approach may be to work with LLMs in a multi-prompt interactive session, but this is ad-hoc, and hence, not generalizable or replicable. Instead, we propose and pilot a **Leave-one-out approach** for refining codebooks, which is systematic, and hence, generalizable and replicable.

2. Related Work

Qualitative analysis has been used extensively in studies in computer science education, to analyze datasets such as semi-structured interviews [6], code interviews [7], CS student reflections [8], online forum discussion [9], think-aloud interviews [10], and computer science assignments [11]. The availability of large amounts of text data for analysis in computer science education research has prompted researchers to investigate leveraging the use of LLMs.

Researchers have studied several approaches for using LLMs to create codebooks. In a single-prompt approach, LLMs are used to generate a complete set of themes without human interaction, and focuses solely on prompting strategies [12, 13, 14, 15].

LLMs have been used to iteratively develop codebooks. For example, Jiang [16] describes an iterative coding process in which both humans and an LLM are asked to identify and justify themes; and disagreements between the two are examined to create the next version of the codebook.

LLMs have been dovetailed with human researchers for creating codebooks. This approach may simply consist of two or more steps: LLM-generated themes followed by human refinement [17, 18] or human-generated themes followed by LLM refinement [17], or multiple steps in which LLM-generated themes are analyzed by humans who refine the prompt for another LLM pass [19]. One study [20] that compared a human-developed codebook to a codebook developed first by an LLM and then refined by humans found that the LLM/human codebook tended to contain broader, higher level categories.

Finally, LLMs have been used at each of the steps humans usually take to create codebooks. In this approach, the focus is on writing and revising LLM prompts for each step. DePaoli [21] used Braun and Clark’s six stages of thematic analysis [22] to generate the *entire* codebook using this approach. Naeem [23] and Wang [24] used the same approach, but with more recent LLM models. Currently, efforts are ongoing to automate codebook generation [25, 26].

Whereas the above approaches are for codebook creation, often involving iterative refinement, we propose an approach to refine an existing codebook so that it can be used with additional/new data. Our proposed Leave-One-Out approach can be used to 1) verify whether a theme belongs in a codebook; 2) determine how to split a theme already in a codebook; 3) discover themes that are not already in the codebook. Some advantages of our approach are that it is a single-prompt approach, and therefore, systematic, and hence, generalizable and replicable.

3. Methodology

In this study, we propose a systematic approach to refine codebooks that leverages LLMs, and supports the analysis of qualitative data in the context of computer science education.

3.1. Leave-One-Out Approach

The following are the steps in the Leave-one-out approach:

1. **Sample data:** Identify a random data set such that all the themes in the codebook are represented.
2. **Prompt creation:** Create a codebook-based prompt for LLM.
3. **Leave-one-out:** For each theme in the codebook:
 - a) **Codebook modification:** Remove the theme from the codebook in the prompt.

- b) **Deductive coding:** Have the LLM deductively code the statements in the sample data using the modified codebook.
- c) **Additional themes:** Ask the LLM to propose additional themes to be considered for inclusion in the codebook based on recurring patterns in the sample data.
- 4. **Collating and clustering additional themes:** Collate the additional themes from all the Leave-one-out runs and cluster them thematically.
- 5. **Categorizing additional themes:** Sort the collated themes into one of the following bins:
 - Themes that must be re-included in the codebook
 - Themes that must be split into two or more sub-themes
 - New themes that must be included in the codebook

Next, we will pilot this approach to analyze the data collected from computing students (subsection 3.2) using an existing codebook (Table 1).

3.2. Context

In order to pilot our proposed approach, we used data previously collected and deductively coded by a team of researchers. The goal of data collection was to study how computing students understood persistence and use this information to help educators design interventions to promote persistence among computing students. Persistence is a key professional disposition in computer science [27]. It is highly valued by computing students [27] and industry [28] and is an important component of ACM curricular recommendations [29]. Therefore, data was collected to identify the academic behaviors that computing students associate with the persistence disposition.

Data was collected during six consecutive terms (2022-2025) in seven undergraduate computing courses at five baccalaureate institutions in the US. The study was approved by the institutional review boards of the participating institutions. A total of 896 statements were collected from 510 students who consented to the use of their data for research.

During data collection, computing students were first provided the definition of persistence as “to stick with a task until it is completed even when the task seems difficult and even when you have doubts about your ability to complete the task.” They were then asked to respond to the following open-ended question after multiple course assignments: “Describe an example of you being persistent when completing this assignment. Otherwise, describe the circumstances that prevented you from being persistent.” Their text responses or statements constituted coding units in the study.

Each statement was treated as one coding unit for the purpose of deductive coding. Researchers did not break up statements into distinct codes before identifying themes because all the statements were individual responses to the same prompt, and were therefore, focused and short. Researchers deductively coded these statements into themes using the codebook in Table 1 developed for the behaviors students associate with the persistent disposition. Note that the codebook contains seven themes of student behaviors, listed in the order that researchers used to code each statement based on the primary meaning in the statement. Two additional themes were used to code statements that did not have enough information to glean meaning (not pertinent) and to code statements in which students indicated they could not be persistent (not applied). Each statement was coded into exactly one theme in Table 1, which was deemed the statement’s *ground-theme*.

In this work-in-progress study, we investigated how the codebook in Table 1 could be refined using the Leave-one-out approach (Section 3.1). We used Claude (Sonnet 4.6 free, web-based) as the LLM because it is known to have fewer hallucinations [30].

3.3. Piloting the Approach

We elaborate the piloting of each step in the Leave-one-out approach below:

1. **Sample data:** We selected a random subset of 200 statements from the original 896 statements and verified that each theme in Table 1 was proportionately represented.

Table 1Codebook of themes for *persistent* disposition

No.	Theme	Description
1	Overcoming challenges	Working through errors or adverse conditions, focusing on challenges that are overcome
2	Aiming at quality	Applying effort with the goal of quality outcome
3	Seeking help	Using resources or asking questions
4	Iterating	Redoing, changing, redesigning, focusing on the process
5	Time on task	Spending a lot of time, extra time or scheduling necessary time on a regular basis
6	Finishing	Completing the task
7	Putting in effort	Keeping at it or not giving up. Fall-through category of behaviors for which none of the above themes applies.
a	Not pertinent	Insufficient information to indicate persistent behaviors
b	Not applied	Factors preventing student from being persistent

2. **Prompt creation:** We created a codebook-based prompt consisting of the following components: persona, task, context, codebook, supporting evidence, beneficiaries, and output (see Appendix).

3. **Leave-one-out:**

- a) **Codebook modification:** For each of the seven behavior themes, we created a *Leave-one-out* version of the prompt with that theme removed in the included codebook. Thus, the codebook in each of the seven prompts contained six behavior themes plus “not pertinent” and “not applied”. The prompt text provided for leaving out *Overcoming Challenges* is shown in the Appendix.
- b) **Deductive coding/Additional themes:** Each of the seven prompts was run as a separate project in Claude to deductively code the sample data and suggest additional themes. The counts of additional themes suggested in each run were respectively: 3, 2, 3, 5, 4, 3, and 3, for a total of 23 overlapping themes in seven runs.

4. **Collating and clustering additional themes:** The additional themes were collated and clustered and are listed below, but without the explanation that was also provided by the LLM for each theme. The suggested themes are followed in parenthesis by the number (from Table 1) of the left-out theme in the original codebook:

- “Emotional regulation / Coping with frustration” (1), “Emotional / motivational regulation” (2), “Emotional regulation / managing frustration” (3), “Working through emotional difficulty / self-regulation” (5), and “Emotional regulation / managing frustration” (7)
- “Working under adverse personal circumstances” (4) and “Overcoming adversity / working through personal hardship” (6)
- “Proactive planning / Time management (positive)” (1), “Managing time strategically / planning ahead” (6), “Managing time / effort allocation” (5), and “Scheduling / pacing work over time” (5)
- “Strategic use of breaks as a persistence tool” (4) and “Strategic pacing / self-regulation” (7)
- “Collaboration / Teamwork as persistence” (1), “Collaborative persistence / team coordination” (2), “Collaborative persistence” (3), “Team/Collaboration Coordination as Persistence” (4), “Team coordination / accountability” (5), “Collaborating / coordinating with team” (6), and “Collaboration / teamwork” (7)
- “Seeking external resources / help-seeking” (3) and “Iterative design revision” (4)
- “Self-directed learning as persistence” (4)

5. **Categorizing additional themes:** As described next in Section 4, we found that Claude recommended:

- Re-inclusion of three themes in the very run in which they had been left out: “Seeking help” (3), “Iterating” (4) and “Time on task” (5);

- Splitting two themes—“Overcoming challenges” and “Time on task” into three sub-themes each;
- Adding “Collaboration” and “Self-directed learning” as two new themes.

4. Findings and Discussion

We present three results from our pilot study as the main takeaways of the proposed Leave-one-out approach.

In the following cases, the LLM identified and recommended re-inclusion of a theme in the very run in which it had been left out:

- “Seeking external resources / help-seeking” when “Seeking help” (3) was omitted;
- “Iterative Design Revision” when “Iterating” theme (4) was omitted;
- Two themes: “Scheduling / pacing work over time” and “Managing time / effort allocation” when “Time on task” was omitted. The two together constitute “Time on task”.

The recommendation for re-inclusion of these themes validates their presence in the codebook.

The LLM recommended splitting the following themes in the original codebook into multiple sub-themes:

- “Overcoming challenges” to be split into emotional, technical and personal sub-themes :
 - **Emotional:** “Emotional regulation”, described variously as a “psychological dimension”, a “metacognitive coping strategy”, and an “affective dimension”, and explained as “managing emotions (frustration, discouragement, mental blocks, fear of failure, shame, self-doubt)” by practices such as “taking breaks, sleeping on a problem, resetting after frustration.” This theme was recommended in five out of seven runs.
 - **Technical:** The LLM suggested that “Overcoming challenges” be restricted to “technical or task”-related obstacles in three runs.
 - **Personal:** In two runs, the LLM proposed “Working under adverse personal circumstances” pertaining to “personal or situational challenges” (versus affective or technical challenges), such as “illness, personal crises, sleep deprivation, and competing obligations - obstacles external to the assignment”.
- “Time on task” to be split into managing, pacing and strategy sub-themes:
 - **Managing:** “Strategically managing time”, explained as “deliberately, proactively planning when and how to work” through activities such as “daily scheduling, early starts, frequent commits” (1 run).
 - **Pacing:** The LLM suggested that “Time on task”, renamed as “Scheduling / pacing work over time”, be restricted to “spending time on task” (3 runs).
 - **Strategy:** The LLM proposed “Strategic pacing/self-regulation” as another “meta-cognitive strategy”-related extract of “Time on task”, explained as “breaking work into manageable sessions, taking intentional breaks to reset, or spacing work over days to stay effective” (2 runs).

Splitting these themes into sub-themes is supported by the fact that the sub-themes were included in the descriptions of the themes in the original codebook.

The LLM recommended the following additional themes:

- “Collaborative/teamwork”, described as a “socially-enacted form” (as compared to individual effort), and explained as “sustained engagement with partners or team members” through activities such as “following up, reminding, checking in, coordinating, keeping team on track, messaging

group members, dividing and verifying work, ensuring role completion” to ensure task completion. This theme is particularly relevant to group assignments. The LLM noted that these statements were coded as “Putting in effort”, the default theme in the original codebook, but “Putting in effort” could be set aside for individually-focused activities. Remarkably, this theme was recommended in all seven runs.

- In one run, the LLM proposed “Self-directed learning as persistence”, dealing with “independent skill acquisition under difficulty” and explained as “teaching themselves new tools, languages, or concepts required to complete the task”. This theme is at the intersection of two dispositions: persistence and self-directedness.

The two themes that were not recommended for re-inclusion, or splitting were “Aiming at high quality” to which only 3 of 200 statements belonged, and “Finishing”, which was conceptually close to the default theme of “Putting in effort” (See Table 1).

Takeaway 1. Leave-one-out approach helped validate the inclusion of some themes in the codebook, helped split some themes into more coherent sub-themes and suggested inclusion of additional themes not previously considered.

In the leave-one-out runs, ideally, the only statements whose coding should change are the ones originally coded with the left-out ground-theme. We tabulated the themes identified for each statement across the seven runs. We calculated the number of times a statement was coded the same as the ground-theme in the seven runs. We define this metric as *statement-stability*. Since there were seven runs, the statement-stability ranged from 0 to 6, with 6 representing the greatest coding consistency. Statements with the highest statement-stability figure could be considered candidates for selection as exemplars of a theme. Such exemplars can be included in the codebook to improve the responses of an LLM since one-shot prompts are known to yield better results than zero-shot prompts [17].

Some statements with a *statement-stability* of 6 for the various themes, which make them ideal candidates for exemplars, are as follows:

- “Overcoming challenges”: “My game files were deleted and I had to start over mid way through the semester, I build the game on my PC and on my laptop.”
- “Aiming at high quality”: “I had to be persistent in making sure the scenarios don’t take too long or become too convoluted. In other words, I had to be persistent in making the scenarios balanced and not too convoluted.”
- “Finishing”: “I completed this assignment on time unlike other times.”
- “Seeking help”: “When I don’t understand a concept, I look for other resources to help me learn the material.”
- “Iterating”: “I think highlighting the game over and over and extensively trying to get a general idea/layout going”
- “Time on task”: “I haven’t moved from my room since I started doing my work for this project today.”
- “Putting in effort”: “When completing this assignment, I showed a trait of persistence by constantly getting in touch with my partner so that we can rationalize the task and make changes.”

Takeaway 2. Leave-one-out approach helped shortlist candidates for selection as exemplars of themes.

Finally, we analyzed the cohesion of each theme in the codebook. We calculated the average statement-stability score for all the statements with the same ground-theme. We refer to this metric as *theme-stability*. The greater the theme-stability, the less likelihood of a statement in this ground-theme being coded differently. Since the maximum statement-stability score is 6, the maximum theme-stability score is also 6.

The themes with the lowest theme-stability scores were:

- “Not pertinent” (1.19, N=16): When a student did not specify how they were persistent (e.g., “persistent when something doesn’t work”), stated how they were persistent filling out the survey instead of the assignment (e.g., “I was persistent when answering these questions.”), talked about themselves instead of the assignment (e.g., “I am persistent with everything I do.”) or talked about some other disposition (e.g., “This was a group assignment so we finished as a class”), researchers coded the statements as “Not pertinent”. The LLM tried to fit such statements into other themes.
- “Putting in effort” (1.91, N=23): This was the default theme used for statements that could not be clearly coded as one of the other themes. The LLM tried to fit these statements into other themes.

The theme-stability scores of the other themes ranged from 3.83 (“Time on task”, N=23) to 5.22 (“Overcoming challenges”, N=59) with “Aiming at high quality” as an outlier (6.0, N=3).

Takeaway 3. Leave-one-out approach helped quantitatively identify stable themes in the codebook.

Our work retains researchers as the primary creators of codebooks while using LLMs as a resource for refining extant codebooks to improve their clarity, independence, and coverage of categories. While LLMs have the potential to contribute to codebook refinement, the final judgment should be left to researchers with domain knowledge.

The changes in the themes obtained through the Leave-one-out approach will help educators design more targeted interventions to promote the essential disposition of persistence [27] among computer science students, e.g., the three sub-themes of “Overcoming challenges” can be addressed differently by instructors: provide more assignment scaffolding for technical, provide flexible deadlines for personal, and provide motivational support for emotional sub-themes.

5. Limitations

We piloted the Leave-one-out approach using one data set and one codebook. In order to establish its generalizability, we plan to test it with additional data, especially data collected by other researchers, and other codebooks.

GenAI has been continually improving in its capabilities. Agentic AI can already be used to automate the entire Leave-one-out approach. Therefore, it is conceivable that future versions of GenAI will obviate the need for Leave-one-out approach.

In the codebook, the themes were listed in priority order, i.e., the themes were listed and were to be considered in a sequential order. It is not clear that the approach will be equally effective for codebooks without a priority order.

We used Claude in our pilot study. Using multiple GenAIs and seeking convergence in their responses across multiple runs with the same data may yield better refinement of the codebook.

Acknowledgments

Partial support for this work was provided by the National Science Foundation under grant DUE-2216121. The authors acknowledge Stephanos Matsumoto, Renée McCauley and Mihaela Sabin for their help in deductively coding the original data set and Devorah Kletenik for her help in data collection.

Declaration on Generative AI

The authors used Gemini to format some of the references.

References

- [1] A. J. Bingham, From data management to actionable findings: A five-phase process of qualitative data analysis, *International Journal of Qualitative Methods* 22 (2023) 1–11.
- [2] S. A. Silverio, J. Hall, J. Sandall, Time and qualitative research: Principles, pitfalls, and perils, in: *The Sixth Annual Qualitative Research Symposium*, 2020.
- [3] L. Hu, Programming and 21st century skill development in K-12 schools: A multidimensional meta-analysis, *Journal of Computer Assisted Learning* 40 (2024) 610–636. doi:<https://doi.org/10.1111/jcal.12904>.
- [4] N. A. Ernst, G. Bavota, AI-driven development is here: Should you worry?, *IEEE Software* 39 (2022) 106–110. doi:[10.1109/MS.2021.3133805](https://doi.org/10.1109/MS.2021.3133805).
- [5] A. S. Hayes, “Conversing” with qualitative data: Enhancing qualitative research through large language models (LLMs), *International Journal of Qualitative Methods* 24 (2025) 16094069251322346. doi:[10.1177/16094069251322346](https://doi.org/10.1177/16094069251322346).
- [6] I. Hou, O. Man, K. Hamilton, S. Muthusekaran, J. Johnykutty, L. Zadeh, S. MacNeil, ‘All roads lead to ChatGPT’: How generative AI is eroding social interactions and student learning communities, Association for Computing Machinery, New York, NY, USA, 2025.
- [7] S. Kannam, Y. Yang, A. Dharm, K. Lin, Code interviews: Design and evaluation of a more authentic assessment for introductory programming assignments, Association for Computing Machinery, New York, NY, USA, 2025.
- [8] M. Celepkolu, K. E. Boyer, Thematic analysis of students’ reflections on pair programming in CS1, in: *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, SIGCSE ’18*, Association for Computing Machinery, New York, NY, USA, 2018, p. 771–776. doi:[10.1145/3159450.3159516](https://doi.org/10.1145/3159450.3159516).
- [9] P. Mulgund, R. Singh, M. Gillenson, Y. Li, C. Oliver, “Not my first choice”: Why software testing struggles to attract early-career professionals, in: *2025 Computers and People Research Conference (SIGMIS-CPR ’25)*, ACM, Waco, TX, USA, 2025. doi:[10.1145/3716489.3728443](https://doi.org/10.1145/3716489.3728443).
- [10] J. Whalley, A. Settle, A. Luxton-Reilly, A think-aloud study of novice debugging, *ACM Trans. Comput. Educ.* 23 (2023). doi:[10.1145/3589004](https://doi.org/10.1145/3589004).
- [11] X. Liu, A. F. Zambrano, R. S. Baker, A. Barany, J. Ocumpaugh, J. Zhang, M. Pankiewicz, N. Nasiar, Z. Wei, Qualitative coding with GPT-4: Where it works better, *Journal of Learning Analytics* 12 (2025) 169–185. doi:[10.18608/jla.2025.8575](https://doi.org/10.18608/jla.2025.8575).
- [12] A. Parfenova, A. Marfurt, J. Pfeffer, A. Denzler, Text annotation via inductive coding: Comparing human experts to LLMs in qualitative data analysis, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 6471–6484. doi:[10.18653/v1/2025.findings-naacl.361](https://doi.org/10.18653/v1/2025.findings-naacl.361).
- [13] J. L. Marquard, C. L. Martin, S. S. N. Theesfeld, A. L. Feldmann, A. C. Trofholz, E. C. Wick, G. B. Melton, Manual vs. AI-assisted qualitative analysis: A methodological comparison of nurse perspectives on surgical recovery, *International Journal of Qualitative Methods* 25 (2026) 1–18. doi:[10.1177/16094069261429593](https://doi.org/10.1177/16094069261429593).
- [14] L. Hamilton, D. Elliott, A. Quick, S. Smith, V. Choplin, Exploring the use of AI in qualitative analysis: A comparative study of guaranteed income data, *International Journal of Qualitative Methods* 22 (2023) 1–13. doi:[10.1177/16094069231201504](https://doi.org/10.1177/16094069231201504).
- [15] X. Liu, J. Zhang, A. Barany, M. Pankiewicz, R. S. Baker, Assessing the potential and limits of large language models in qualitative coding, in: Y. J. Kim, Z. Swiecki (Eds.), *Advances in Quantitative Ethnography*, Springer Nature Switzerland, Cham, 2024, pp. 89–103. doi:[10.1007/978-3-031-76335-9_7](https://doi.org/10.1007/978-3-031-76335-9_7).
- [16] H. Jiang, S. Corrigan, K. Peppler, T. Islam Erana, Bridging human and machine perspectives: Integrating large language models into collaborative coding and peer debriefing for qualitative inquiry, in: *Joint Proceedings of LAK 2025 Workshops*, volume 3995, CEUR Workshop Proceedings, 2025. URL: <https://ceur-ws.org/Vol-3995/>, co-located with the 15th International Conference on

Learning Analytics and Knowledge (LAK 2025), Dublin, Ireland, March 03–07, 2025.

- [17] A. Barany, N. Nasiar, C. Porter, A. F. Zambrano, A. L. Andres, D. Bright, M. Shah, X. Liu, S. Gao, J. Zhang, S. Mehta, J. Choi, C. Giordano, R. S. Baker, ChatGPT for education research: Exploring the potential of large language models for qualitative codebook development, in: A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, I. I. Bittencourt (Eds.), *Artificial Intelligence in Education*, Springer Nature Switzerland, 2024, pp. 134–149. doi:10.1007/978-3-031-64299-9_10.
- [18] A. Karimov, M. Saarela, X. Liu, Z. Wei, A. F. Zambrano, A. Barany, R. S. Baker, J. Ocumpaugh, S. Mammadova, T. Kärkkäinen, ChatGPT-assisted codebook design for learning analytics datasets in multiple languages: A case study, in: *Advances in Quantitative Ethnography. International Conference on Quantitative Ethnography*, volume 2677, Springer, 2025, pp. 177–192. doi:10.1007/978-3-032-12229-2_12.
- [19] R. Sinha, I. Solola, H. Nguyen, H. Swanson, L. Lawrence, The role of generative AI in qualitative research: GPT-4’s contributions to a grounded theory analysis, in: *Symposium on Learning, Design and Technology (LDT ’24)*, ACM, New York, NY, USA, 2024, p. 17–25. doi:10.1145/3663433.3663456.
- [20] X. Liu, Z. Wei, A. Barany, J. Ocumpaugh, R. S. Baker, A. F. Zambrano, Y. Zhou, C. Giordano, Exploring differences between hybrid GPT-human and human-created qualitative codebooks in an educational game, in: *Advances in Quantitative Ethnography*, Springer Nature Switzerland, 2026, pp. 193–208. doi:10.1007/978-3-032-12229-2_13.
- [21] S. De Paoli, Performing an inductive thematic analysis of semi-structured interviews with a large language model: An exploration and provocation on the limits of the approach, *Social Science Computer Review* 42 (2024) 997–1019. doi:10.1177/08944393231220483.
- [22] V. Braun, V. Clarke, Using thematic analysis in psychology, *Qualitative Research in Psychology* 3 (2006) 77–101. doi:10.1191/1478088706qp0630a.
- [23] M. Naeem, T. Smith, L. Thomas, Thematic analysis and artificial intelligence: A step-by-step process for using ChatGPT in thematic analysis, *International Journal of Qualitative Methods* 24 (2025) 1–18. doi:10.1177/16094069251333886.
- [24] Q. Wang, M. Erqsous, K. E. Barner, M. L. Mauriello, LATA: A pilot study on LLM-assisted thematic analysis of online social network data generation experiences, *Proceedings of the ACM on Human-Computer Interaction* 9 (2025) 1–28. doi:10.1145/3711022.
- [25] A. Katz, G. C. Fleming, J. B. Main, Thematic analysis with open-source generative AI and machine learning: A new method for inductive qualitative codebook development, *Humanities and Social Sciences Communications* 13 (2026). doi:10.1057/s41599-026-06508-5.
- [26] J. Gao, K. T. W. Choo, J. Cao, R. K.-W. Lee, S. Perrault, CoAICoder: Examining the effectiveness of AI-assisted human-to-human collaboration in qualitative analysis, *ACM Transactions on Computer-Human Interaction (TOCHI)* 31 (2023) 1–38. doi:10.1145/3617362.
- [27] A. N. Kumar, R. K. Raj, S. G. Aly, M. D. Anderson, B. A. Becker, R. L. Blumenthal, E. Eaton, S. L. Epstein, M. Goldweber, P. Jalote, D. Lea, M. Oudshoorn, M. Pias, S. Reiser, C. Servin, R. Simha, T. Winters, Q. Xiang, *Computer Science Curricula 2023*, ACM, 2024. doi:10.1145/3712204.
- [28] D. Tagare, S. Janakiraman, M. Exter, S. Duan, M. Sabin, J. Tavakoli, Dispositions that computing professionals value in the workplace: Systematic literature review and interviews with professionals, in: *Proc. of the 2023 ACM Conference on International Computing Education Research*, ACM, 2023, p. 270–283. doi:10.1145/3568813.3600118.
- [29] A. Clear, A. Parrish, J. Impagliazzo, P. Wang, P. Ciancarini, E. Cuadros-Vargas, S. Frezza, J. Gal-Ezer, A. Pears, S. Takada, H. Topi, G. van der Veer, A. Vichare, L. Waguespack, M. Zhang, *Computing Curricula 2020 (CC2020): Paradigms for Future Computing Curricula*, Technical Report, Association for Computing Machinery and IEEE Computer Society, New York, NY, USA, 2020. URL: <https://dl.acm.org/citation.cfm?id=3467967>.
- [30] S. Rubinstein, A. Mohsin, R. Banerjee, W. Ma, S. Mishra, M. Kwok, P. Yang, J. L. Warner, A. J. Cowan, Summarizing clinical evidence utilizing large language models for cancer treatments: A blinded comparative analysis, *Frontiers in Digital Health* 7 (2025) 1569554. doi:10.3389/fdgth.2025.1569554.

Appendix: Prompt - Leave Out *Overcoming Challenges*

Persona: You are a thematic analysis expert.

Task: Please analyze the open-ended statements written by undergraduate students in computing courses to the question: “Describe an example of you being persistent when completing this assignment. Otherwise, describe the circumstances that prevented you from being persistent.”

Beneficiary: Your analysis will be used by computing education researchers to summarize how students understand persistence in undergraduate computing courses.

Context: For analysis purposes, Persistence is defined as “to stick with a task until it is completed even when the task seems difficult and even when you have doubts about your ability to complete the task.”

Codebook: The following are the themes identified for persistence, along with their explanations:

1. “Aiming at high quality” theme is explained as “Applying effort with the goal of a high quality outcome”
2. “Seeking help” theme is explained as “using resources” or “asking questions”
3. “Iterating” theme is explained as “Redoing, changing, redesigning (focusing on the process)”
4. “Time on task” theme is explained as “spending a lot of time (or extra time)” or “scheduling necessary time on a regular basis”
5. “Finishing” theme is explained as “completing the task”
6. “Putting in effort” theme is explained as “kept at it” or “did not give up”. This theme applies if the enumerated statement is about persistence, but none of the above themes apply.

The themes are listed in the order in which they should be considered for thematic analysis. The statements written by students are listed at the end in an enumerated list. Please identify the themes of persistence in the enumerated statements. When doing so, ignore the word persistent in the statement unless the statement is about not being persistent.

When listing the themes identified in each enumerated statement:

- If an enumerated statement contains more than one theme, list all the themes. But, list the dominant theme first. The dominant theme is what the student conveys as the primary example of being persistent.
- If an enumerated statement does not contain any theme, list “not pertinent”.
- If an enumerated statement explains why persistence was not applied, list “not applied”.

Supporting evidence: For each theme, list the words or phrases in the statement that led you to identify the theme in the statement. If the theme is “not pertinent” or “not applied”, list the reason for the classification.

Output: Print the results in csv format with entries in the following order: the statement number, the dominant theme (if any), the words or phrases that identify the theme in the statement, the other themes, and for each, the words or phrases that identify that theme.

Next, print the statistics of the number of statements in each theme, and the number of statements with multiple themes.

Finally, suggest any other themes not listed above, if any, that you found recurring in the statements that should be considered,

Here are the enumerated statements: <coding-units included in prompt, but omitted for this paper>