International Society of the Learning Sciences

Processes Matter: How ML/GAI Approaches Could Support Open Qualitative Coding of Online Discourse Datasets

John Chen, Alexandros Lotsos, Grace Wang, Lexie Zhao, Bruce Sherin, Uri Wilensky, Michael Horn
civitas@u.northwestern.edu, alexandroslotsos2026@u.northwestern.edu, xinyuezhao2020@u.northwestern.edu, GraceWang2025@u.northwestern.edu, bsherin@northwestern.edu, uri@northwestern.edu, michael-horn@northwestern.edu
Northwestern University

Abstract: Open coding, a key inductive step in qualitative research, discovers and constructs concepts from human datasets. However, capturing extensive and nuanced aspects or "coding moments" can be challenging, especially with large discourse datasets. While some studies explore machine learning (ML)/Generative AI (GAI)'s potential for open coding, few evaluation studies exist. We compare open coding results by five recently published ML/GAI approaches and four human coders, using a dataset of online chat messages around a mobile learning software. Our systematic analysis reveals ML/GAI approaches' strengths and weaknesses, uncovering the complementary potential between humans and AI. Line-by-line AI approaches effectively identify content-based codes, while humans excel in interpreting conversational dynamics. We discussed how embedded analytical processes could shape the results of ML/GAI approaches. Instead of replacing humans in open coding, researchers should integrate AI with and according to their analytical processes, e.g., as parallel co-coders.

Introduction

Qualitative coding is the process of systematically identifying, generating, and organizing concepts from data. Open coding, a key first step in qualitative coding, aims to inductively discover and construct concepts from a dataset while remaining "open" and "as exhaustive as possible" (Corbin & Strauss, 2008; Fereday & Muir-Cochrane, 2006). CSCL researchers have used machine learning (ML) methods for analyzing discourse data for deductive analysis (Stahl, 2015) and open coding (Lopez-Fierro & Nguyen, 2024; Sinha et al., 2024). However, lacking a ground truth, open coding is more challenging to support and evaluate. Even though recent studies show promising results (Lopez-Fierro & Nguyen, 2024; Zambrano et al., 2023), it remains unclear how AI should be best integrated into open coding processes.

We conducted a systematic study on an online community dataset to evaluate the strengths and weaknesses of five ML/GAI open coding approaches. Due to page constraints, please refer to our full version (recommended as a CSCL long paper) for more literature review and study details. Comparing results from four human coders and five ML/GAI approaches, our initial analysis shows strengths of item-level (i.e., line-by-line) approaches over topic modeling or chunk-level approaches in identifying nuanced open codes. Further analysis shows that while machine coders could identify the majority of human codes, they excelled at contributing codes that were grounded in message contents, as opposed to those grounded in conversational dynamics.

Analytical processes are essential for BOTH human and machine coders to produce high-quality open coding outcomes. We explained much of the machine coders' performances through the analytical processes embedded in their prompts. We suggest 1) researchers should integrate appropriate ML/GAI approaches by matching them with the contextual needs of analytical processes, 2) better ML/GAI approaches for qualitative research may be developed by integrating human coding processes, and 3) instead of replacing humans in the workflow, qualitative researchers should consider using ML/GAI approaches as, and only as, parallel co-coders.

Related work

ML and GAI techniques have been extensively studied by the CSCL community to support qualitative analysis, particularly for large discourse datasets, yet past studies have mostly focused on deductive analysis (Zheng et al., 2019; Erkens & Janssen, 2008). For open coding, the number and nature of underlying codes are unknown before the analysis, necessitating the study of generation approaches. Some recent examples include:

1. *Topic modeling* (Baumer et al., 2020), where resulting topics help researchers focus on key data patterns. Despite efforts, the difficulty in interpreting and evaluating the results limits its power (Grootendorst, 2022; Sievert & Shirley, 2014).
2. *GAI models* (De Paoli, 2023; Sinha et al., 2024), where researchers iteratively provide data pieces with relevant instructions (e.g., research questions, coding instructions) to elicit codes from the model. However, GAI models can still miss nuance and produce vague themes (De Paoli, 2023; Hamilton et al.,

2023), prompting recent work to advocate for transparency in researcher-AI collaboration (Lopez-Fierro & Nguyen, 2024) and careful prompting strategies (Byun et al., 2024; Sinha et al., 2024), with few attempting to compare open coding results from different ML/GAI strategies.

While metrics such as intercoder reliability (de Araujo et al., 2023) or precision and recall (Rosé et al., 2008) work well for deductive coding, open coding has no "ground truth" for reference. Without a viable alternative, many papers had to rely on them for evaluating open codes (Gao et al., 2023; Gebreegziabher et al., 2023; Parfenova, 2024; Rietz & Maedche, 2021). To address this, Zhao et al. (2024) proposed using semantic similarity to match between machine and human open codes. However, the approach indirectly positions human codes as a "ground truth" and underutilizes GAI's potential in identifying novel insights - which recent CSCL studies have highlighted (e.g., Lopez-Fierro & Nguyen, 2024; Zambrano et al., 2023). Building on team-based approaches (Cascio et al., 2019), our recent work proposes a computational approach to measure open codes *from multiple coders* (Chen, Lotsos, Zhao, Hullman, et al., 2024). Yet, while the approach can identify likely "novel" codes, it is still unclear how machine coders can or cannot contribute to the open qualitative analysis.

Empirical analysis and results

To understand ML/GAI's capability in identifying emergent insights from online discourses, we examined a text-based discourse dataset, coded by four human coders and five automated coding approaches (Chen, Lotsos, Zhao, Wang, et al., 2024). Our analysis is done in four stages: 1) a first-pass evaluation, which identifies two approaches as closest to human coders and narrows down the analysis; 2) systematic merging to identify open codes uniquely covered by human or machine coders; 3) identifying each code's potential contribution to further analysis; 4) interpreting each code's grounding of contribution in relation with the raw data. For a more detailed version of ML/GAI coding approaches and our analysis process, please refer to the full version.

The discourse dataset is from the online chat channel of Physics Lab, a mobile learning platform for youths to construct interactive physics simulations and share their projects. The research question explores "how an online community emerged in Physics Lab." The dataset (with 127 messages between designers and teachers) was independently open-coded by four human coders with various degrees of contextual knowledge and coding experience. Using the five ML/GAI coding approaches with GPT-4o-0513 (Chen, Lotsos, Zhao, Wang, et al., 2024), Machine coders generated the open codes with assigned roles and tasks, such as "*You are an expert in thematic analysis with grounded theory, working on open coding*." They were also provided with the research question and relevant background information in their prompts, with no codebooks or examples beforehand.

Table 1

Codes for "Mechanics will Have to Wait until Electromagnetism is Figured Out; it will Take Some More Time."

Human Coders	future update, managing expectations, explanation of upcoming features, vague on responses to the feature request, preview of update, respond to feature request
BERTopic + LLM	feature prioritization
Chunk Level	future plans
Chunk Level, Structured	*N/A (not identified as part of any code)*
Item-Level	development timeline, feature prioritization, subject specific tools, user feedback
Item-Level, Verb Phrases	manage user expectations, explain current focus, set timeline for mechanics experiments

Table 1 provides example codes of a designer's response to a teacher's request. Both *Topic Modeling + LLM* and *chunk-level* approaches identified broad themes such as "feature prioritization" or "future plans," yet, their codes lack the nuanced aspects of fine-grained human codes. *Item-level approaches* are more capable of finding fine-grained open codes that match human codes, such as "*manage user expectations*" or "*explain current focus*". Moreover, codes such as "*set timeline for mechanics experiments*" summarize the message's content and contribute to the idea of "timeline," with the potential to complement human analyzers. Therefore, the rest of our analysis focuses on item-level approaches.

Table 2

An Overview of Each Coding Approach's Potential Contribution to Further Analysis.

	Total	4 Human Coders	Item-Level, Both Approaches	Item-Level	Item-Level, Verb Phrases
# Uniquely Covered	57	17	10	10	20
- Little Gain	25	7	8	3	7
- Minor Gain	12	2	2	5	3
- Substantial Gain	**20**	**8**	**0**	**2**	**10**

Since machine and human coders often use different labels for the same concept (e.g., "manage user expectations" vs. "managing expectations"), we systematically merged human (340 codes) and machine (522 codes) codes through human-AI collaboration (Chen, Lotsos, Zhao, Hullman, et al., 2024). Two researchers analyzed 81 randomly ordered merged codes: a coding approach is considered to *cover the merged code* only if any of AI's suggested codes covers 1) the same idea or 2) a narrower scope than the merged code. Two researchers independently coded the same codes *without seeing the algorithm's decisions* for three rounds. After each round, they calculated the inter-coder reliability and fully reconciled the discrepancies. The Cohen's Kappa between R1 and the algorithm is 0.78 (substantial); between R2 and the algorithm, 0.56 (moderate); and between the consensus and the algorithm, *0.68 (substantial).* The first row of Table 2 demonstrates our results.

With the merged codes, two researchers analyze each code and coder's potential contribution to the analysis around *Physics Lab's community formation*. The core idea is the unique contribution of each code: if human coders did not identify the code during open coding, were we missing potential insights from the dataset?

We categorized each code into *Little, Minor*, and *Substantial* Gain, as shown in Table 2. A significant portion of uniquely covered codes (25 out of 57, 44%) has *little potential to contribute (Little Gain)*. In most cases (20), this is due to another coder identifying similar or more narrow ideas. For example, both AI coders identified the code "*community context*" from the message "Mainly, the school is building an information-based school." None of the 4 human coders applied an equivalent code. Yet, two coders did each apply a pair of codes, "school needs" and "context," that can capture an equivalent idea in combination. Another 12 codes are deemed "*Minor Gain*": 1) Some codes are too specific, such as "Augmented Reality" and "Multi-language Support" from a version update note posted by the designer. Given the research's goal around community formation, the researchers judged that such codes are less likely to help. 2) Some codes' contributions overlap with others. For example, the item-level verb phrase approach identified "acknowledge provided resources" in a user response: "I saw the group files, thank you." Meanwhile, human codes "acknowledgement" or "sending resources" cover a similar conceptual space. Missing the code may lead to potential limited (thus minor) loss.

To reveal human and machine coders' relative strengths and weaknesses, our final analysis focuses on the 20 *Substantial Gain* codes identified above. Two researchers examined each code independently and wrote analytical memos before coding them together. We report the most prominent feature of the analysis: the *source*, or grounding, of each coder's *Substantial Gain* codes (Table 3). In short, machine coders excel at identifying codes from the *content* of messages, while human coders are better at *conversational dynamics*.

Table 3
Sources of Substantial Gain Codes for Each Coding Approach.

	Total	4 Human Coders	Item-Level, Both Approaches	Item-Level	Item-Level, Verb Phrases
# Substantial Gain	20	8	0	2	10
- From Content	13	2	0	2	9
- From Conversational Dynamics	7	6	0	0	1

Some codes are grounded in the *content* of the messages it was applied to, where "content" here is narrowly construed. For example, consider this designer message: "Consulting the teachers in the group: which type of intersection is used in the circuit diagrams in the current textbooks?*"* The item-level verb phrase approach applied a code "*consult on educational standards*." Here, the code is dependent on a narrow focus on the specific subject, "current textbook," as present in the message. Yet, instead of focusing on the subject alone, the machine coder interpreted it as "*educational standards*," a *conversation topic* that can contribute to forming research questions. Perhaps teachers' participation in the chat channel was in part triggered by discussions about educational standards, contributing to community formation.

Some codes, though grounded in the narrow content of messages, provide further interpretation beyond the immediate content. For example, the item-level verb phrase approach found the code "*align with educational standards*" from the user message "Yes, the common one is still the old style", which answers the designer's question in the last paragraph. The code's core contribution is the higher-level action of "*aligning*." Essentially, the machine coder interprets the message content: by affirming "the old style" to be "the common one," the user was "*aligning*" their response to the designer - and the software's design - with educational standards.

Other codes are grounded in *conversational dynamics*. By the term, we refer to the form and function of the message within an ongoing conversation rather than its content. Take a simple example: As long as a message serves as a response to a previous message, it can be coded as a "response." In our dataset, right before the start of Table 1's conversation, the designer explained the features of circuit diagrams. Then, a user said, "Can you also include mechanics experiments?*"* A human coder coded this *incident* as "*topic change without response*." The code was inferred from the place and role of the message in the conversation: a user initiated the change of

topic without responding to a prior message. If further analysis reveals many similar topic changes, researchers may ask follow-up questions: Is it an indicator where users gained initiative in the conversation? Or is it a disruptive action that would drive other users away?

Codes grounded in conversational dynamics can provide further interpretation as well. For example, after the designer replied, "*Mechanics will have to wait until electromagnetism is figured out; it will take some more time," a user wrote: "Don't aim for completeness, it should be categorized and refined one by one*." By applying the code "*understanding designer's situation*," a human coder not only identifies the situation - but also points to the idea of "*understanding*" between community members that is worthy of further exploration.

Discussions and conclusions

Processes embedded in ML/GAI coding approaches

To maximize ML/GAI's potential for analyzing discourse datasets, we must understand the analytical processes embedded in different ML/GAI coding approaches and match them with the contextual needs of human analytical processes. Even though all were supplied with the same information (RQ, context, and instructions), we identified major differences: *Item-level approaches* identify fine-grained codes more aligned with grounded theory expectations, while *BERTopic and chunk-level approaches* identify broader codes akin to themes.

Such differences are largely explained by the underlying mechanical differences of the five approaches. While BERTopic leverages generative AI models, it remains a clustering method rooted in semantic similarity, which both enables and limits its effectiveness. It performs well when examples are semantically close but struggles with semantically orthogonal or context-dependent expressions like sarcasm or minimal responses ("yeah"). Similarly, chunk-level coding approaches often generate fewer and more thematic "open codes," likely influenced by post-training processes geared toward human preferences that include shorter responses. The problem is particularly salient in open coding, where the number of codes is inherently unpredictable. Human coders face a similar issue, often defaulting to thematic codes when analyzing larger data chunks (Gibbs, 2007). On the other hand, line-by-line coding promotes analytic precision and closeness to the data—a principle we found helpful for machine coders, too. Item-level approaches, especially when combined with instructions to use verb phrases, encouraged models to generate more grounded and nuanced codes such as "seek confirmation" or "express frustration with current limitations."

This shows the profound impact of embedded processes on ML/GAI approaches. Rather than arguing that item-level approaches are somehow better than chunk-level approaches or topic modeling, we argue that each approach has a place based on its embedded process, and researchers should adopt appropriate ML/GAI approaches *according to their human analytic processes*.

Towards human-AI collaboration in open coding processes

Our central contribution is a process for evaluating the strengths and weaknesses of machine coders in open coding relative to human coders. By carefully collaborating with a computational algorithm for code merging, we systematically identified which codes were shared or uniquely contributed by each codebook and categorized their grounding to reveal patterns in coder strengths. Applied to an online discourse dataset common in CSCL research, machine coders using line-by-line approaches performed impressively, recovering most human-identified codes and even contributing additional, potentially insightful ones. However, they struggled to generate truly novel codes, particularly those grounded in conversational dynamics. Our process offers a path forward for rigorously evaluating and enhancing human-AI collaboration in qualitative analysis.

Our study shows the potential of ML/GAI approaches in assisting open coding of discourse datasets, commonly studied in CSCL studies (Stahl, 2015). Acknowledging that GAI models are evolving quickly, *for now, we suggest researchers use ML/GAI approaches only as parallel co-coders in the open coding process*. Open coding is about more than just producing codes and researchers **must** become acquainted with the data, likely requiring that they engage in some open coding themselves if they are to produce research and findings that advance the field of CSCL (Corbin & Strauss, 2008). As parallel co-coders, ML/GAI approaches can contribute to a more complete picture of ideas, complementing humans' strengths in higher-level interpretations. Even so, more critical thoughts and broader evaluations are needed to maximize ML/GAI's potential and minimize the risks in open coding.

References

Baumer, E. P. S., Siedel, D., McDonnell, L., Zhong, J., Sittikul, P., & McGee, M. (2020). Topicalizer: Reframing core concepts in machine learning visualization by co-designing for interpretivist scholarship.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*.

Byun, C., Vasicek, P., & Seppi, K. (2024, May 8). Chain of Thought Prompting for Large Language Model-driven

Qualitative Analysis. *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. LLMs as Research Tools: CHI 2024.

Cascio, M. A., Lee, E., Vaudrin, N., & Freedman, D. A. (2019). A Team-based Approach to Open Coding: Considerations for Creating Intercoder Consensus. *Field Methods*, *31*(2), 116–130.

Chen, J., Lotsos, A., Zhao, L., Hullman, J., Sherin, B., Wilensky, U., & Horn, M. (2024). *A Computational Method for Measuring "Open Codes" in Qualitative Analysis*. *arXiv Preprint.*

Chen, J., Lotsos, A., Zhao, L., Wang, G., Wilensky, U., Sherin, B., & Horn, M. (2024). *Prompts Matter: Comparing ML/GAI Approaches for Generating Inductive Qualitative Coding Results*. *arXiv Preprint*.

Corbin, J., & Strauss, A. (2008). Chapter 10 / Analyzing Data for Concepts. In *Basics of Qualitative Research (3rd ed.): Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, Inc.

Davis, N. R., Vossoughi, S., & Smith, J. F. (2020). Learning from below: A micro-ethnographic account of children's self-determination as sociopolitical and intellectual action. *Learning, Culture and Social Interaction*, *24*, 100373.

de Araujo, A., Papadopoulos, P. M., McKenney, S., & de Jong, T. (2023). Automated coding of student chats, a trans-topic and language approach. *Computers and Education: Artificial Intelligence*, *4*, 100123.

De Paoli, S. (2023). Performing an Inductive Thematic Analysis of Semi-Structured Interviews With a Large Language Model: An Exploration and Provocation on the Limits of the Approach. Social Science Computer Review, 0(0), 1–23.

Erkens, G., & Janssen, J. (2008). Automatic coding of dialogue acts in collaboration protocols. *International Journal of Computer-Supported Collaborative Learning*, *3*(4), 447–470.

Fereday, J., & Muir-Cochrane, E. (2006). Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development. *International Journal of Qualitative Methods*, *5*(1), 80–92.

Gao, J., Choo, K. T. W., Cao, J., Lee, R. K.-W., & Perrault, S. (2023). CoAIcoder: Examining the Effectiveness of AI-assisted Human-to-Human Collaboration in Qualitative Analysis. *ACM Transactions on Computer-Human Interaction*, *31*(1), 6:1-6:38.

Gebreegziabher, S. A., Zhang, Z., Tang, X., Meng, Y., Glassman, E. L., & Li, T. J.-J. (2023). PaTAT: Human-AI Collaborative Qualitative Coding with Explainable Interactive Rule Synthesis. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–19.

Gibbs, G. R. (2007). Thematic coding and categorizing. *Analyzing Qualitative Data*, *703*(38–56).

Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv Preprint arXiv:2203.05794*.

Hamilton, L., Elliott, D., Quick, A., Smith, S., & Choplin, V. (2023). Exploring the Use of AI in Qualitative Analysis: A Comparative Study of Guaranteed Income Data. *International Journal of Qualitative Methods*, *22*, 16094069231201504.

Lopez-Fierro, S., & Nguyen, H. (2024). *Making Human-AI Contributions Transparent in Qualitative Coding*. *Proceedings of ISLS Annual Meetings (CSCL)*.

Parfenova, A. (2024). Automating Qualitative Data Analysis with Large Language Models. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics.*

Rietz, T., & Maedche, A. (2021). Cody: An AI-Based System to Semi-Automate Coding for Qualitative Research. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–14.

Rosé, C., Wang, Y.-C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., & Fischer, F. (2008). Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International Journal of CSCL*.

Sievert, C., & Shirley, K. (2014). LDAvis: A method for visualizing and interpreting topics. *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, 63–70.

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

Stahl, G. (2015). A decade of CSCL. *International Journal of Computer-Supported Collaborative Learning*.

Zambrano, A. F., Liu, X., Barany, A., Baker, R. S., Kim, J., & Nasiar, N. (2023). From nCoder to ChatGPT: From Automated Coding to Refining Human Coding. *Advances in Quantitative Ethnography*.

Zhao, F., Yu, F., & Shang, Y. (2024). A New Method Supporting Qualitative Data Analysis Through Prompt Generation for Inductive Coding. *2024 IEEE International Conference on Information Reuse and Integration for Data Science (IRI)*, 164–169.

Zheng, J., Xing, W., & Zhu, G. (2019). Examining sequential patterns of self-and socially shared regulation of STEM learning in a CSCL environment. *Computers & Education*, *136*, 34–48.