

When Humans and LLMs Disagree: Understanding Human-LLM Disagreement in Learner Intent Annotation for Educational Chatbot Dialogues

Isabel Schorr^{1,2}, Lisa Bardach^{3,4}, Haoxiang Gao¹, Enkelejda Kasneci^{1,2}

¹ Technical University of Munich (TUM), Chair for Human-Centered Technologies for Learning

² Munich Center for Machine Learning (MCML)

³ Justus Liebig University Giessen, Department of Psychology

⁴ Kaunas University of Technology

{isabel.schorr, haoxiang.gao, enkelejda.kasneci}@tum.de

lisa.bardach@psychol.uni-giessen.de

Abstract

As large language model (LLM) chatbots become embedded in education, researchers need scalable ways to analyze what learners are actually doing in student-chatbot interactions. To address this challenge, we propose a learning theory-informed annotation framework for learner intent and engagement in educational chatbot dialogues, operationalizing solution seeking, knowledge seeking, cognitive engagement, and mental model expression through a hierarchical decision process, and apply it manually to a dataset of 2,393 learner-chatbot interactions (408 double-coded). We then evaluate two LLMs as zero-shot annotators and compare human-human, LLM-LLM, and human-LLM agreement across both intermediate decisions and final intent labels. While the two LLMs exhibit agreement patterns approaching human-human reliability on the final label (human $\kappa = 0.73$ vs. LLM $\kappa = 0.67$), their agreement with human reference labels is substantially lower ($\kappa \leq 0.53$), indicating that automated annotation errors cannot be understood from aggregate scores alone. Through exploratory qualitative coding of disagreement cases, we find that divergences can arise from ambiguous learner intent, competing valid interpretations, and human annotation variability, rather than model failure alone.

1 Introduction

Large language model (LLM) chatbots are increasingly used in education (Baidoo-Anu and Ansah, 2023; Kasneci et al., 2023; Li and Zheng, 2026). Whereas much research on students' use of LLM chatbots has relied on self-reported perceptions and experiences (e.g., Chan and Hu, 2023; Morell-Mengual et al., 2025), recent work has begun to examine authentic learner-chatbot interactions using annotated conversation data. Understanding intent-aware dialogue acts in these interactions may support learner behavior analysis, intent-aware chatbot design, and more effective use of educational

chatbots. However, annotated datasets for student-chatbot interactions remain scarce, and manual annotation is difficult to scale. Reliable automated annotation could substantially reduce this bottleneck, yet its performance in educational chatbot settings remains underexplored. To address these gaps, we develop a dialogue annotation scheme for student-chatbot conversations, manually annotate a corpus of learner utterances from such interactions, and evaluate the reliability of zero-shot LLM-based automated annotation. We further examine disagreement patterns between human and LLM annotations to identify strengths and limitations of automated labeling in educational chatbot contexts.

Our code and supporting data used for automated annotation are available at: <https://github.com/d-isasterhub/studychat-automatic-labeling>.

2 Related Work

Our work intersects research on educational dialogue systems, dialogue annotation, learning theory, and automated annotation.

2.1 Dialogue Act Annotation and Discourse Modeling

Conversational interactions are central to different lines of research, including educational dialogue and intelligent tutoring systems (Graesser et al., 2004; Litman and Silliman, 2004; Dzikovska et al., 2010; Li and Zheng, 2026). Dialogue acts structure conversations, and substantial work has focused on annotating and modeling dialogue through structured representations of communicative function (Shriberg et al., 2004; Li et al., 2017; Suresh et al., 2022; Karan et al., 2023; Hu et al., 2026). Foundational frameworks such as DAMSL, the Switchboard Dialogue Act Corpus (SwDA), and the ISO 24617-2 standard define taxonomies of dialogue acts that capture the functional role of ut-

terances (e.g., questions, statements, acknowledgments) across conversational contexts (Allen and Core, 1997; Jurafsky et al., 1997; Bunt et al., 2010, 2012, 2016, 2020). Dialogue acts have been used as intermediate signals to improve conversational modeling, for example, in next-utterance prediction tasks, where they can guide response selection (Kumar et al., 2018; Srivastava et al., 2023). However, traditional dialogue act schemes are typically designed for human-human interaction and often lack representations of learning-relevant behaviors. For example, under ISO 24617-2 or SwDA, the utterance "Can you give me the solution?" and "Does a for-loop repeat until the condition becomes false?" can both be annotated as questions, despite reflecting fundamentally different learning behaviors: the former seeks a solution, whereas the latter seeks conceptual understanding or confirmation. Consequently, educational settings require labels that capture learning-relevant intentions and behaviors. In our work, we draw on Learning-to-Learn (L2L) as the conceptual foundation for defining learning-relevant dialogue acts.

2.2 Learning-to-Learn (L2L)

L2L, a key competence for lifelong learning (European Parliament and Council of the European Union, 2006; Council of the European Union, 2018), encompasses regulating, monitoring, and reflecting on one's learning, drawing on cognition, metacognition, and self-regulated learning (SRL) (Flavell, 1979; Winne and Hadwin, 1998; Zimmerman, 2002; Schorr et al., 2026). Recent advances in conversational AI suggest that chatbots can support but also disrupt learning processes central to L2L, including information retrieval, problem-solving, and critical or creative thinking (Kasneji et al., 2023; Darvishi et al., 2024; Zhang et al., 2024).

Learning theories such as L2L provide rich conceptual accounts of learning but are not readily operationalized for fine-grained annotation of learner-chatbot dialogues. Operationalizing these theories would enable systematic analyses of learner-chatbot interactions, such as Epistemic Network Analysis of coded dialogue (Shaffer et al., 2016). While L2L provides the theoretical basis for our characterization of learning behaviors, recent work has proposed a range of approaches for analyzing learner-chatbot interactions.

2.3 Analyzing Learner Behavior in Student-Chatbot Interactions

A growing body of work examines how students engage with educational chatbots, using self-reported instruments (Hou et al., 2025; Wu et al., 2026; Dai et al., 2026) or, more directly, interaction logs: Lai (2024) and Lyu and Ding (2025) map each student prompt onto one of four SRL phases. For instance, Lyu and Ding (2025) may code both a request for a finished paragraph and a request for concept clarification as "control," since both occur during task execution rather than at the planning or reflection stages. Jin et al. (2026) characterize reliance by crossing how actively a student asks for help with how actively they process the chatbot's response. Domain-specific coding schemes have been proposed for chatbot use cases, including essay revision (Han et al., 2024) and programming (Ghimire and Edwards, 2024). Our work complements these approaches in two ways. First, rather than organizing utterances by learning phase (Lai, 2024; Lyu and Ding, 2025), manner of engagement (Jin et al., 2026), or domain-specific communicative function (Han et al., 2024; Ghimire and Edwards, 2024), we contribute a hierarchical, utterance-level annotation scheme centered on the solution- versus knowledge-seeking distinction that is intended to generalize across task domains. While Lyu and Ding (2025) offer a comparable fine-grained taxonomy of 16 microlevel actions, it is organized around the SRL phases and instantiated as a fixed-prompt-action dictionary validated on the writing-support domain. Our scheme directly targets the offloading-versus-understanding distinction, is validated in a programming-coursework setting, and resolves category overlap through an explicit, prioritized decision order rather than dictionary matching. Second, unlike Lai (2024), Lyu and Ding (2025), Jin et al. (2026), and Han et al. (2024), whose schemes are applied through manual coding and used for downstream pattern discovery or classifier training, we focus on the reliability of the scheme itself: we assess whether it can be applied consistently by both humans and zero-shot LLMs and explore the sources of human-LLM disagreement.

2.4 Natural Language Processing (NLP) for Dialogue and LLM-Based Annotation

Recent work has explored LLMs for annotation across diverse domains (Wang et al., 2024; Nasu-

tion and Onan, 2024; Movva et al., 2024; Ahmed et al., 2025; Qamar et al., 2025; Ahtisham et al., 2026). However, LLM annotation performance is highly task-dependent (He et al., 2025; Panagakos and Wolken, 2025): while models can achieve strong zero- and few-shot results in some classification settings (Anglin and Ventura, 2025; Chae and Davidson, 2026), complex annotation remains challenging and may benefit from hybrid human-LLM workflows or multi-step annotation pipelines (Ostyakova et al., 2023; Labruna et al., 2023; Kim et al., 2024; Qamar et al., 2025). Existing work typically evaluates LLM annotations against human labels using precision, recall, and F1. However, human annotations themselves may contain inconsistencies or annotator bias (Thomas et al., 2022; Haliburton et al., 2024, 2025). Therefore, rather than treating disagreements solely as model failures, we explore them to identify limitations in annotation schemes and data.

3 Task

Building on this literature, we contribute a hierarchical, theory-informed, domain-general intent scheme and evaluate its reliability across both human and zero-shot LLM annotation, a question prior schemes have not addressed. Given a learner-chatbot exchange represented as a sequence of (prompt, response) pairs, our task is to assign structured labels that capture theory-informed, pedagogically meaningful interactional moves. To this end, we develop a coding framework grounded in L2L theory and refined through iterative, bottom-up analysis of real learner-chatbot interaction data. Formally, given a dialogue context $D = \{(u_1, a_1), \dots, (u_n, a_n)\}$, where u_i denotes a learner utterance and a_i the corresponding chatbot response, the task is to assign one interactional label $y_i \in \mathcal{Y}$ for each exchange (u_i, a_i) .

Our task definition integrates three key objectives:

- **Conceptual structuring:** We investigate whether interactional learning processes in learner-chatbot dialogues can be decomposed into a set of clearly distinguishable interaction moves or learner intents grounded in theory-driven and empirically derived categories.
- **Human reliability:** We assess whether these interactional categories can be reliably annotated by human annotators, thereby evaluating

the interpretability and stability of the proposed coding framework.

- **Computational modeling:** We examine the extent to which these labels can be automatically predicted using zero-shot LLMs, and analyze systematic failure cases where model predictions diverge from human annotations.

We use chatbot interaction data from a university-level Computer Science course, where annotation requires integrating multiple sources of context, including the assignment setting, the surrounding chat history, and domain-specific knowledge. In addition, the learning theory-informed annotation scheme makes the interpretation of interactional behaviors non-trivial for both human and automatic annotation.

4 Data

Chatbot interaction data. We use the StudyChat dataset (McNichols et al., 2026), which consists of real-world student interactions with a ChatGPT-like assistant collected in an undergraduate artificial intelligence course at the University of Massachusetts Amherst. The dataset is released under the Creative Commons Attribution 4.0 International (CC BY 4.0) license,¹ with our use of it being consistent with these licensing conditions. The dataset spans multiple assignments and contains multi-turn, English-language dialogues between students and the assistant. Importantly, the dataset captures authentic help-seeking behavior in a naturalistic educational setting, making it well-suited for observing student natural interaction patterns. McNichols et al. (2026) collected the dataset under IRB approval from consenting participants, with anonymization and content filtering applied before release. We restrict our study to the Spring 2025 (s25) subset of the dataset to maintain a consistent course context and assignment structure.

Assignment files. The dataset creators provide supplementary materials on the Hugging Face dataset page.² These materials include the original assignment files, which are highly relevant to our annotation task. Each assignment consists of 2 to 8 files, including Python source files, Jupyter notebooks, Markdown files, CSV files, and plain text files.

¹<https://creativecommons.org/licenses/by/4.0/>

²<https://huggingface.co/datasets/wmcnichols/StudyChat>

Assignment summaries. To facilitate both human annotation and downstream computational processing, the first author created summaries for each assignment. These summaries provide a high-level overview of the task objectives, required concepts, and expected outputs, allowing annotators to quickly understand the context without inspecting all underlying files. From a computational perspective, the summaries offer a compact representation of each assignment that can reduce input length and highlight the most salient information.

5 Annotation Scheme

While the original dataset provides dialogue act annotations that capture the functional role of each message, these labels are not intended to distinguish how learners engage with the learning task (e.g., whether they seek solutions or understanding). With L2L as a conceptual lens, we therefore developed an annotation scheme that infers learners’ solution- versus knowledge-seeking intent and captures behavioral proxies of their cognitive and metacognitive engagement.

In line with the standard in qualitative and dialogue annotation work (Saldaña, 2025), we developed the annotation scheme in two steps:

1. Codebook development: The first author constructed an initial codebook using a hybrid approach, combining deductive (theory-driven) codes with inductive refinements emerging from examination of the data.
2. Codebook refinement: Multiple annotators iteratively refined the initial codebook through pilot coding, followed by systematic comparison of coding decisions, discussion of disagreements, and revision of code definitions and boundaries.

Appendix A provides more details about the annotation scheme development phases.

During the two phases, the annotation scheme evolved from an initially independent set of labels to a structured decision-tree annotation process that derives a single final annotation label (Figure 1). Besides the main label derived from the decision tree, the annotation task included two additional aspects: confirming or correcting incorrect assignment mappings (a1-a7) and determining whether the prompt is related to an assignment task (related, unrelated, or setting up the learning environment).

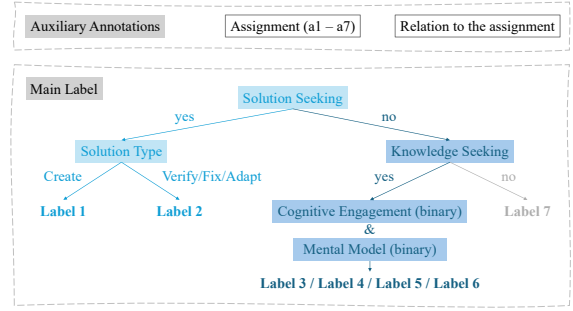


Figure 1: Hierarchical decision process for label assignment. Internal nodes represent sequential decisions, and each leaf corresponds to a final main label. Auxiliary annotations ("Assignment" and "Relation to the assignment") are attached as additional attributes.

The main label is derived using the following deliberations within the decision tree (Figure 1):

- **Solution Seeking:** Is the learner’s intent to get a working solution for an assignment (sub)task?
- **Solution Type:** Does the learner request a new solution or want to verify/fix/adapt an existing one?
- **Knowledge Seeking:** Does the learner seek information, knowledge, or understanding?
- **Cognitive Engagement:** Is there evidence of active cognitive engagement with the previous message(s)?
- **Mental Model:** Does the learner show evidence of their current mental model (assumptions, explanations, or understanding)?

When *solution seeking* and *knowledge seeking* are both plausible, an utterance is labeled *solution seeking* whenever the requested response would directly produce or complete an assignment task or subtask, or would require only minor adaptation to do so. Figure 2 illustrates the application of the annotation scheme to an example interaction. Appendix B provides a more detailed version of the final annotation scheme with representative examples.

6 Human Annotation

The following section delineates the manual labeling process and the inter-rater agreement between the two labelers.

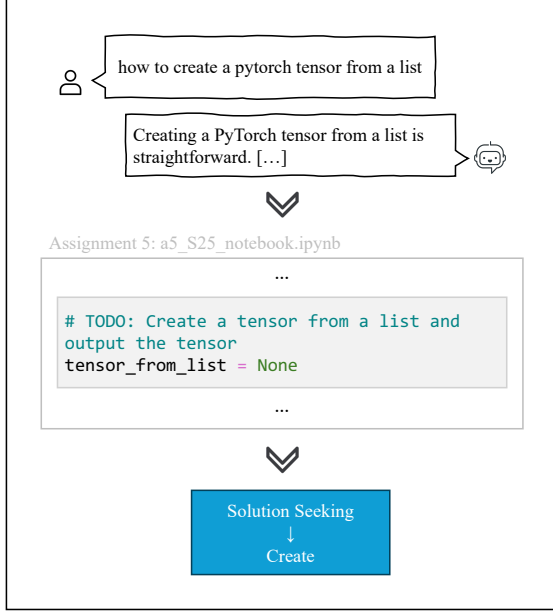


Figure 2: Example annotation of a student utterance in a student-chatbot interaction adapted from the StudyChat dataset (McNichols et al., 2026), using the dialogue annotation scheme introduced in this work.

6.1 Procedure

We employed two annotators of the same research group for the manual labeling process: the first author, with a background in machine learning and tutoring, and a graduate research assistant. Both participated in the iterative codebook development process described above, and no separate formal training was conducted. During annotation, they had full access to all materials for the seven assignments provided by the original dataset (including all associated files), along with summaries of each assignment prepared by the first author. For each annotation instance, annotators labeled prompt-response pairs while working through and retaining access to the full conversational context of the chat. Besides prompt, response, and label fields, each instance included the assignment identifier provided by the dataset (e.g., "a7"), as well as the top five embedding-based similarity scores between the prompt and assignment segments across all assignments, which supported the identification of relevant task connections. The annotators' tasks included correcting incorrect assignment mappings when necessary, determining the relationship to the assignment, and rendering a decision for the main label following the decision tree.

We organized the annotation process into weekly phases. In Phases 1 and 2, both annotators worked

Sprint	N	κ
Phase 1 (Codebook refinement)	72	0.55
Phase 2 (Codebook refinement)	69	0.76
Phase 3	117	0.77
Phase 4	80	0.71
Phase 5	70	0.78

Table 1: Cohen's κ scores for the main label across the five human annotation rounds using the finalized codebook.

independently on the full set of arbitrarily selected chats. In Phases 3-5, chats were split between annotators by workload, with a randomly selected 10% subset per phase double-coded. This double-coding allowed the annotators to monitor inter-rater consistency and maintain a stable interpretation of the coding scheme throughout the annotation process. At the end of each week, the annotators met to review inter-rater agreement, discuss disagreements, and produce a consolidated version of the double-annotated data, representing the final human reference label used for human-LLM evaluation unless otherwise stated.

Over a five-week annotation period, two annotators labeled 2,393 student utterances from the full dataset of 9,709 utterances, including 408 double-coded and 1,985 single-coded instances. We concluded annotation after five weeks because inter-rater agreement stabilized across annotation phases, suggesting that the annotators reached stable agreement patterns across phases.

6.2 Agreement

Table 1 summarizes the agreement for the main label of both human labelers across phases, all of which use the final codebook version, including the two final rounds of the refinement phase and three annotation phases using the frozen scheme. Besides the first phase, all phases maintain a stable inter-rater agreement score above 0.71, considered to be substantial (Landis and Koch, 1977). The lower agreement in the first phase using the final codebook version likely reflects the ongoing alignment of annotators' shared interpretation of the labels.

7 LLM-based Annotation

In this section, we delineate our approach for automatic, zero-shot labeling of the StudyChat dataset using our annotation scheme.

7.1 General Setup

We employ two LLMs from the GPT family, namely GPT-5.4 and GPT-OSS:120B, using the OpenAI API and local inference, respectively. We use temperature 0.0 for inference. We annotate all human-coded instances with GPT-OSS:120B and the double-coded subset with GPT-5.4. At the chat level, each conversation is labeled with its most likely corresponding assignment. At the utterance level, each student prompt is labeled with (1) its relation to the assignment and (2) a main label. The latter is determined following our hierarchical decision process, resulting in a sequence of 2-4 structured decisions per student utterance. Available details on hardware/software configuration, runtimes, token usage, and estimated costs are reported in Appendix C.

7.2 Prompting Setup

To reduce manual prompt engineering, we implement our annotation pipeline using DSPy signatures and modularize the task into structured sub-decisions (Khattab et al., 2024). The full set of signatures is provided in Appendix D. Due to the hierarchical and context-dependent nature of our labeling scheme, prompts are dynamically constructed with different context components depending on the annotation objective and the respective node in the decision process. Appendix E details the contextual information provided to the LLMs for each labeling decision.

7.3 Retrieval Setup

To provide assignment-specific context, we implement an embedding-based retrieval pipeline over assignment materials. We preprocess assignment files by chunking them according to file type (see Appendix F).

All resulting chunks are embedded using all-MiniLM-L6-v2 and stored in ChromaDB along with metadata linking each vector to its source text and assignment. At inference time, student utterances are encoded with the same embedding model and used as a query. We perform similarity search using the default squared L2 distance and retrieve the top- k most relevant chunks, restricted to the assignment associated with the current chat. These retrieved chunks are then incorporated into the prompt for context-sensitive labeling decisions.

7.4 Chat History Construction

To control context length, we construct truncated chat histories, as the annotation prompts already include extensive annotation instructions, label definitions, and (in some cases) retrieved assignment chunks. The exact history representation depends on the labeling objective. We provide details in Appendix G regarding which parts of the chat history we include for each annotation decision.

8 Results

This section compares agreement across human-human, LLM-LLM, and human-LLM settings measured in terms of Cohen’s kappa and accuracy to compare annotation consistency without treating human labels as absolute ground truth. To ensure comparability, all agreement measures are computed on the double-coded instances. Human-human agreement uses the two independent human annotations before consolidation, while human-LLM agreement compares LLM annotations against the corresponding consolidated human annotations. We additionally evaluate human-LLM agreement on single-coded instances to assess agreement stability and representativeness. The number of compared instances N per variable may vary depending on the position in the decision tree, as comparisons are only made when both annotators reach the same node.

8.1 Human-Human Agreement

We observe generally high inter-annotator agreement between human annotations (Table 2). Frequent discussions between the two annotators regarding label definitions and disagreement cases likely contributed to this consistency and helped establish a shared understanding of the annotation scheme. However, the annotators showed comparably low agreement on *cognitive engagement* and *mental model*. These dimensions require subjective judgments about implicit cognitive processes and learner understanding, which proved difficult to operationalize consistently and may require additional discussion to align interpretations.

8.2 LLM-LLM Agreement

The two annotation LLMs achieve agreement levels overall comparable to human-human agreement (Table 3). However, their agreement pattern differs from that of the human annotators: the models show relatively strong agreement on *cognitive en-*

Variable	N	κ (Acc.)
assignment	408	0.91 (0.93)
assignment relation	408	0.70 (0.98)
solution seeking	408	0.79 (0.91)
solution type	272	0.85 (0.93)
knowledge seeking	100	0.85 (0.96)
cognitive engagement	82	0.73 (0.87)
mental model	82	0.52 (0.89)
main label	408	0.73 (0.80)

Table 2: Detailed human-human agreement across annotation dimensions.

Variable	N	κ (Acc.)
assignment	408	0.99 (1.00)
assignment relation	408	0.71 (0.97)
solution seeking	408	0.53 (0.78)
solution type	217	0.52 (0.76)
knowledge seeking	103	0.89 (0.97)
cognitive engagement	85	0.78 (0.93)
mental model	85	0.74 (0.91)
main label	408	0.67 (0.78)

Table 3: Detailed agreement between annotation LLMs across annotation dimensions.

gement and *mental model*, but lower agreement on *solution seeking* and *solution type*.

8.3 Human-LLM Agreement

Human-LLM agreement is lowest among all pairings, but improves when using GPT-5.4 compared to GPT-OSS:120B, which aligns with differences in model scale and training/alignment maturity (Table 4). Additional analyses of label distributions and per-class metrics (Appendix H) indicate that disagreement is concentrated in low-support learner-intent categories, particularly in *mental model* "true" paths, helping explain why accuracy can remain comparatively high despite lower Cohen’s κ . The results likely reflect complementary observations: humans are more consistent in *solution seeking* and *solution type*, while LLMs show stronger alignment in *cognitive engagement* and *mental model*. Lower human-LLM agreement for *cognitive engagement* and *mental model* is consistent with lower human-human agreement, suggesting these dimensions require judgments about less directly observable learner states. We analyze these annotation differences in the following section.

To assess whether double-coded instances disproportionately contribute to lower human-LLM

agreement due to higher ambiguity, we additionally evaluate agreement on the subset of single-coded examples using GPT-OSS:120B. The agreement remains largely stable for the main label (double-coded: $\kappa = 0.41$, single-coded: $\kappa = 0.43$), suggesting that the observed disagreement reflects systematic differences in how humans and LLMs interpret and classify the exchanges.

9 Analysis of Disagreement

In this section, we analyze disagreements between human and GPT-5.4 annotations through an iterative qualitative coding process. We focus on cases in which the consolidated human annotation and GPT-5.4 prediction diverge on at least one decision in the hierarchical labeling process. This analysis is intended to distinguish model limitations from other sources of disagreement. The first annotator derived the categories through iterative comparison of disagreement cases, the second annotator reviewed the definitions and assignments, and no independent reliability coefficient was computed for this secondary coding. This process resulted in three broad areas of disagreement: model limitations, task and annotation design issues, and data ambiguity and human factors. Because disagreements can arise from multiple factors, we allowed instances to receive more than one disagreement code where appropriate. The distribution of these disagreement types is summarized in Figure 3. The coding is exploratory, intended to identify recurring sources of human-LLM disagreement and contextualize the quantitative agreement results. Future research should examine the reproducibility and generalizability of the taxonomy through independent annotation studies. We briefly introduce the categories below, while Appendix I provides detailed descriptions and examples.

9.1 Model Limitations

We identify several error patterns attributable to model limitations, where predictions diverge from reasonable interpretations despite sufficient input information.

- **Context ignored.** The model fails to incorporate relevant contextual information, in particular task excerpts.
- **Reasoning failure.** The model produces decisions that lack coherent justification or contradict explicit cues in the input.

Variable	$N_{gpt_{5.4}}$	$\kappa_{gpt_{5.4}} (Acc.)$	$N_{gpt_{120b}}$	$\kappa_{gpt_{120b}} (Acc.)$
assignment	408	0.96 (0.97)	408	0.97 (0.98)
assignment relation	408	0.47 (0.96)	408	0.48 (0.95)
solution seeking	408	0.63 (0.87)	408	0.50 (0.77)
solution type	248	0.67 (0.83)	221	0.56 (0.79)
knowledge seeking	95	0.38 (0.87)	94	0.55 (0.87)
cognitive engagement	78	0.53 (0.79)	72	0.39 (0.75)
mental model	78	0.48 (0.85)	72	0.37 (0.82)
main label	408	0.52 (0.64)	408	0.41 (0.56)

Table 4: Detailed agreement between human annotators and LLMs across annotation dimensions.

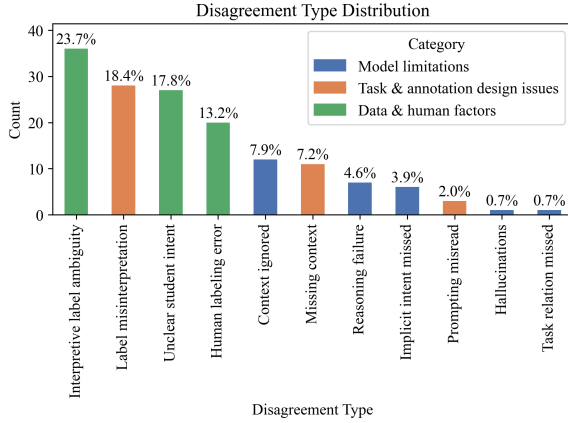


Figure 3: Distribution of disagreement types (percentages indicate share of assigned labels, not instances).

- **Implicit intent missed.** The model tends to over-rely on surface-level phrasing, failing to infer implicit learner intent.
- **Hallucinations.** The model introduces information not present in the dialogue context, attributing decisions to nonexistent prior references.
- **Task relation missed.** The model overgeneralizes topical similarity, treating prompts as assignment-related even when they fall outside the actual course scope.

9.2 Task and Annotation Design Issues

We further identify discrepancies arising from the formulation of the annotation task itself. These issues include both conceptual limitations of the annotation scheme (e.g., underspecified label definitions) and implementation-level limitations of the annotation pipeline (e.g., context handling). While the latter can be addressed through improvements to the LLM-based annotation pipeline, the former require refinements to the annotation framework.

- **Label misinterpretation.** The model exhibits deviations in how label definitions are supposed to be understood.
- **Missing context.** Errors arise from insufficient or misaligned contextual information.
- **Prompting misread.** The model misinterprets the annotation prompt itself, treating meta-instructions as part of the learner utterance.

9.3 Data Ambiguity and Human Factors

Finally, we observe disagreement cases that arise from the inherent ambiguity of the data and variability in human annotation, rather than from model or task design limitations.

- **Interpretive label ambiguity.** Disagreements arise when instances permit multiple plausible interpretations, allowing both human and model labels to be reasonably justified.
- **Unclear student intent.** Disagreements arise when the student’s intent is ambiguous, making it difficult to determine the goal of the interaction.
- **Human labeling error.** Variability and errors in human annotation lead to disagreement.

9.4 Observations

Our results, shown in Figure 3, indicate that many disagreements may not be explained solely by model error: 23.7% reflect interpretive label ambiguity, 17.8% involve unclear intent, and 13.2% involve human error, highlighting the substantial role of data- and human-related factors in disagreement patterns. This pattern is particularly evident in *solution seeking* disagreements, where ambiguous intent and competing valid interpretations frequently

contribute to disagreement, while *knowledge seeking* disagreements often involve human annotation errors (see Appendix J).

10 Discussion

Across our agreement analyses, high agreement between the two LLMs did not translate into equivalent agreement with human annotations, a relevant consideration when using LLMs as annotators. A central contribution of our work is the observation that annotation disagreement between humans and LLMs may not indicate model failure alone. In complex, context-dependent annotation tasks such as learning-related dialogue, disagreements may stem from ambiguity, underspecified learner intent, or inconsistency in human annotation. This observation aligns with prior work highlighting the limited reliability of human annotation and the inherent complexity of dialogue annotation. The observed agreement with human labels should be interpreted diagnostically rather than as evidence that LLMs can autonomously replace human annotators. The disagreements highlight situations in which human oversight remains important and motivate the use of LLMs as decision-support tools.

From the perspective of automatic labeling, our findings indicate that neither human nor LLM annotations consistently constitute reliable ground truth, and that, in some cases, no unambiguous ground truth may exist at all. Both humans and LLMs exhibit complementary strengths and failure modes. This finding suggests that annotation using learning-theory-informed interactional-proxy labels for student-chatbot interactions should be treated less as a strictly human-centered process and more as a form of multi-source supervision, where human and model judgments jointly contribute to label interpretation (e.g., Kim et al., 2024; Wang et al., 2024; Schroeder et al., 2025). Such approaches may reduce annotation effort while helping identify underspecified definitions, ambiguous labels, and missing contextual information. More broadly, both human and LLM annotations may be better understood as weak supervision signals rather than absolute ground truth (cf. Ratner et al., 2017).

Our observations can motivate concrete adjustments for human-LLM annotation pipelines in educational dialogue. Because disagreement concentrates in low-support categories such as *mental model* (Table 8 & 9, Appendix H), predictions

in these regions could be routed to expert review. Since a substantial share of disagreements reflects genuine ambiguity rather than error (Figure 2), decision nodes may allow both human and automated annotators to flag underspecified cases instead of forcing a choice. Finally, periodic auditing of disagreement patterns, distinguishing codebook drift, missing contextual information, and systematic model errors, could help maintain annotation quality over time and signal when the codebook itself needs revision. These are examples of how the weak-supervision framing above could be translated into concrete pipeline design choices.

Another key insight of our study is the prevalence of unclear user intent, which has direct implications for the design of AI-supported learning systems. From an L2L perspective, unclear intent may reflect both communication-related and metacognitive challenges (cf. Schorr et al., 2026). Learners may struggle not only to formulate their questions precisely, but also to identify what they actually do not understand or need help with. Educational systems should therefore support learners in articulating and refining their goals, rather than assuming intentions behind requests. This finding aligns with prior work emphasizing the importance of metacognitive support and SRL in education and educational systems (e.g., Biggs, 1988; Roll et al., 2007; Dever et al., 2023).

11 Conclusion

In this work, we develop and apply a dialogue annotation scheme for student-chatbot interactions that captures cognitively relevant behaviors such as *solution* and *knowledge seeking*. We evaluate the reliability of manual and automated annotation, using two zero-shot LLMs, and analyze human-model disagreement patterns. Our findings suggest that educational chatbot interactions tend to be ambiguous and difficult to annotate consistently, and that high LLM agreement does not necessarily imply high agreement with human annotations. Rather than viewing such disagreements solely as model failures, our analyses motivate treating human and LLM annotations as complementary supervision signals, with LLMs providing scalable support while human oversight remains essential. The prevalence of unclear learner intent as a main source of annotation disagreement further suggests that AI systems for learning should help learners articulate and refine their requests.

Limitations

Our work is based on a single dataset of student-chatbot interactions and may not fully generalize to other educational contexts, platforms, or age groups. In addition, although our annotation scheme is theory-motivated, it represents one possible operationalization of inherently ambiguous constructs such as intent and knowledge-seeking, which may not always admit a single correct interpretation.

Furthermore, our analysis of automated annotation is limited to two LLMs and does not explore the full range of model families or prompting strategies. Finally, while we include double annotation for a subset of the data, scalability constraints limit the extent of human annotation, reflecting a broader trade-off between annotation quality, coverage, and feasibility in educational dialogue analysis.

We intentionally restrict the present study to zero-shot annotation and disagreement analysis rather than optimizing annotation performance through prompt engineering or retrieval design. As a result, we do not include context ablations, few-shot prompting experiments, or human-in-the-loop simulations, as these would address a different research question focused on improving annotation reliability rather than understanding the sources of human-LLM disagreement. We view such experiments as a promising direction for future work.

Ethical Considerations

Our work involves automatically labeling educational chatbot dialogue with LLMs. A key risk is that such annotations may be treated as objective ground truth, even though learning-related dialogue can be ambiguous and highly dependent on context. Our findings illustrate that differences between human and LLM labels may indicate various valid interpretations rather than clear errors. We therefore position both human and LLM annotations as weak supervision signals rather than absolute ground truth.

Acknowledgments

We used ChatGPT to assist in shortening assignment summaries used as contextual information during annotation. The generated shortened summaries were manually reviewed and verified by the authors to ensure that they accurately reflected the original summaries and preserved all information relevant to the annotation process. No AI-generated

modifications were incorporated without human verification.

Isabel Schorr is supported by the Munich Data Science Institute (MDSI) at the Technical University of Munich (TUM) via a doctoral fellowship. Lisa Bardach is supported by a Jacobs Foundation Research Fellowship, an Elite Program for Postdocs Fellowship from the Baden-Württemberg Foundation, and research funding No. P-EDU-23-28, co-funded by the European Union (the project "Breakthrough in Educational Research" No. 10-044-P-0001) under the 1st April 2025 Agreement with the Research Council of Lithuania (RCOL, LMTLT) and the 14th April 2025 Joint Activity Agreement with Kaunas University of Technology.

Data Availability

The annotated corpus supporting this study is available from the authors upon reasonable request. No openly released version of the annotated corpus is currently available. Consequently, direct reproduction of the annotation results and independent assessment of annotation consistency are limited. To support transparency and facilitate future replication, we provide a detailed description of the annotation methodology, annotation guidelines, and evaluation procedures used in this study.

References

- Toufique Ahmed, Premkumar Devanbu, Christoph Treude, and Michael Pradel. 2025. [Can LLMs replace manual annotation of software engineering artifacts?](#) In *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*, pages 526–538.
- Bakhtawar Ahtisham, Kirk Vanacore, and Rene F Kizilcec. 2026. [Optimizing LLM annotation of classroom discourse through multi-agent orchestration.](#) *arXiv preprint arXiv:2603.13353*.
- James Allen and Mark Core. 1997. [Draft of DAMSL: Dialog act markup in several layers.](#)
- Kylie L Anglin and Claudia Ventura. 2025. [Automatic text classification with large language models: A review of OpenAI for zero- and few-shot classification.](#) *Journal of Educational and Behavioral Statistics*, 50(6):1060–1082.
- David Baidoo-Anu and Leticia Owusu Ansah. 2023. [Education in the era of generative artificial intelligence \(AI\): Understanding the potential benefits of ChatGPT in promoting teaching and learning.](#) *Journal of AI*, 7(1):52–62.

- John Biggs. 1988. [The role of metacognition in enhancing learning](#). *Australian Journal of Education*, 32(2):127–138.
- Harry Bunt, Jan Alexandersson, Jean Carletta, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Kiyong Lee, Volha Petukhova, Andrei Popescu-Belis, Laurent Romary, Claudia Soria, and David Traum. 2010. [Towards an ISO standard for dialogue act annotation](#). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 2548–2555.
- Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex C Fang, Koiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David Traum. 2012. [ISO 24617-2: A semantically-based standard for dialogue annotation](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 430–437.
- Harry Bunt, Volha Petukhova, Emer Gilmartin, Catherine Pelachaud, Alex Fang, Simon Keizer, and Laurent Prévot. 2020. [The ISO standard for dialogue act annotation](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 549–558.
- Harry Bunt, Volha Petukhova, David Traum, and Jan Alexandersson. 2016. [Dialogue act annotation with the ISO 24617-2 standard](#). In *Multimodal interaction with W3C standards: Toward natural user interfaces to everything*, pages 109–135. Springer.
- Youngjin Chae and Thomas Davidson. 2026. [Large language models for text classification: From zero-shot learning to instruction-tuning](#). *Sociological Methods & Research*, 55(2):501–567.
- Cecilia Ka Yuk Chan and Wenjie Hu. 2023. [Students' voices on generative AI: Perceptions, benefits, and challenges in higher education](#). *International Journal of Educational Technology in Higher Education*, 20:43.
- Council of the European Union. 2018. [Council recommendation of 22 may 2018 on key competences for lifelong learning](#). *Official Journal of the European Union*, 189:1–13.
- Yun Dai, Suyu Liu, Sihan Zhou, Sichen Lai, Ang Liu, and Cher Ping Lim. 2026. [Redefining and measuring student agency in AI-assisted learning: Development and validation of the agentic engagement with AI \(AE-AI\) scale](#). *Computers & Education*, 253:105687.
- Ali Darvishi, Hassan Khosravi, Shazia Sadiq, Dragan Gašević, and George Siemens. 2024. [Impact of AI assistance on student agency](#). *Computers & Education*, 210:104967.
- Daryn A Dever, Nathan A Sonnenfeld, Megan D Wiedbusch, S Grace Schmorrow, Mary Jean Amon, and Roger Azevedo. 2023. [A complex systems approach to analyzing pedagogical agents' scaffolding of self-regulated learning within an intelligent tutoring system](#). *Metacognition and Learning*, 18:659–691.
- Myroslava O Dzikovska, Johanna D Moore, Natalie Steinhäuser, Gwendolyn Campbell, Elaine Farrow, and Charles B Callaway. 2010. [Beetle II: A system for tutoring and computational linguistics experimentation](#). In *Proceedings of the ACL 2010 System Demonstrations*, pages 13–18.
- European Parliament and Council of the European Union. 2006. [Recommendation of the European Parliament and of the council of 18 december 2006 on key competences for lifelong learning](#). *Official Journal of the European Union*, 394:10–18.
- John H Flavell. 1979. [Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry](#). *American Psychologist*, 34(10):906–911.
- Aashish Ghimire and John Edwards. 2024. [Coding with AI: How are tools like ChatGPT being used by students in foundational programming courses](#). In *AIED 2024: Artificial Intelligence in Education*, volume 14830 of *Lecture Notes in Computer Science*, pages 259–267. Springer.
- Arthur C Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M Louwerse. 2004. [AutoTutor: A tutor with dialogue in natural language](#). *Behavior Research Methods, Instruments, & Computers*, 36:180–192.
- Luke Haliburton, Sinksar Ghebremedhin, Robin Welsch, Albrecht Schmidt, and Sven Mayer. 2024. [Investigating labeler bias in face annotation for machine learning](#). In *HAI 2024: Hybrid Human AI Systems for the Social Good*, volume 386 of *Frontiers in Artificial Intelligence and Applications*, pages 145–161. IOS Press.
- Luke Haliburton, Jan Leusmann, Robin Welsch, Sinksar Ghebremedhin, Petros Isaakidis, Albrecht Schmidt, and Sven Mayer. 2025. [Uncovering labeler bias in machine learning annotation tasks](#). *AI and Ethics*, 5:2515–2528.
- Jieun Han, Haneul Yoo, Junho Myung, Minsun Kim, Tak Yeon Lee, So-Yeon Ahn, and Alice Oh. 2024. [RECIPE4U: Student-ChatGPT interaction dataset in EFL writing education](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13666–13676.
- Jiaman He, Zikang Leng, Dana McKay, Damiano Spina, and Johanne R Trippas. 2025. [Can we hide machines in the crowd? quantifying equivalence in LLM-in-the-loop annotation tasks](#). In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 426–436.

- Chenyu Hou, Gaoxia Zhu, Vidya Sudarshan, Fun Siong Lim, and Yew Soon Ong. 2025. [Measuring undergraduate students' reliance on generative AI during problem-solving: Scale development and validation](#). *Computers & Education*, 234:105329.
- Zhongtian Hu, Changhong Jiang, Yiwen Cui, Ronghan Li, Wei Zhang, and Jiashi Lin. 2026. [Ubuntu-IntAct: A dialogue-act annotated multi-party conversation dataset and graph-based intention modeling](#). *Information Processing & Management*, 63(5):104696.
- Hyoungwook Jin, Minju Yoo, Jieun Han, Zixin Chen, So-Yeon Ahn, and Xu Wang. 2026. [RelianceScope: An analytical framework for examining students' reliance on generative AI chatbots in problem solving](#). In *Proceedings of the Thirteenth ACM Conference on Learning@ Scale*, pages 136–147.
- Dan Jurafsky, Elizabeth Shriberg, and Debra Binasca. 1997. [Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13](#). Technical report, University of Colorado, Boulder Institute of Cognitive Science.
- Vanja M Karan, Prashant Khare, Ravi Shekhar, Stephen McQuistin, Ignacio Castro, Gareth Tyson, Colin Perkins, Patrick Healey, and Matthew Purver. 2023. [LEDA: a large-organization email-based decision-dialogue-act analysis dataset](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6080–6089.
- Enkelejda Kasneci, Kathrin Seßler, Stefan Küchermann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, and 4 others. 2023. [ChatGPT for good? on opportunities and challenges of large language models for education](#). *Learning and Individual Differences*, 103:102274.
- Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. [DSPy: Compiling declarative language model calls into self-improving pipelines](#). In *The Twelfth International Conference on Learning Representations*.
- Hannah Kim, Kushan Mitra, Rafael Li Chen, Sajjadur Rahman, and Dan Zhang. 2024. [Meganno+: A human-LLM collaborative annotation system](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–176.
- Harshit Kumar, Arvind Agarwal, and Sachindra Joshi. 2018. [Dialogue-act-driven conversation model: An experimental study](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1246–1256.
- Tiziano Labruna, Sofia Brenna, Andrea Zaninello, and Bernardo Magnini. 2023. [Unraveling ChatGPT: A critical analysis of AI-generated goal-oriented dialogues and annotations](#). In *AIxIA 2023: Advances in Artificial Intelligence*, volume 14318 of *Lecture Notes in Computer Science*, pages 151–171. Springer.
- Joel Weijia Lai. 2024. [Adapting self-regulated learning in an age of generative artificial intelligence chatbots](#). *Future Internet*, 16(6):218.
- J Richard Landis and Gary G Koch. 1977. [The measurement of observer agreement for categorical data](#). *Biometrics*, 33(1):159–174.
- Shan Li and Juan Zheng. 2026. [A scoping review of large language model-based pedagogical agents](#). *arXiv preprint arXiv:2604.12253*.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. [DailyDialog: A manually labelled multi-turn dialogue dataset](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.
- Diane Litman and Scott Silliman. 2004. [ITSPOKE: An intelligent tutoring spoken dialogue system](#). In *Demonstration Papers at HLT-NAACL 2004*, pages 5–8.
- Yilin Lyu and Ren Ding. 2025. [Discovering self-regulated learning patterns in chatbot-powered education environment](#). *arXiv preprint arXiv:2510.01275*.
- Hunter McNichols, Fareya Ikram, and Andrew Lan. 2026. [The StudyChat dataset: Analyzing student dialogues with ChatGPT in an artificial intelligence course](#). In *Proceedings of the LAK26: 16th International Learning Analytics and Knowledge Conference*, pages 53–63.
- Vicente Morell-Mengual, Olga Fernández-García, Carmen Berenguer, Jessica Ortega-Barón, María Dolores Gil-Llario, and Verónica Estruch-García. 2025. [Characteristics, motivations and attitudes of students using ChatGPT and other language model-based chatbots in higher education](#). *Education and Information Technologies*, 30:22257–22274.
- Rajiv Movva, Pang Wei Koh, and Emma Pierson. 2024. [Annotation alignment: Comparing LLM and human annotations of conversational safety](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9048–9062.
- Arbi Haza Nasution and Aytuğ Onan. 2024. [ChatGPT label: Comparing the quality of human-generated and LLM-generated annotations in low-resource language NLP tasks](#). *IEEE Access*, 12:71876–71900.
- Lidiia Ostyakova, Veronika Smilga, Kseniia Petukhova, Maria Molchanova, and Daniel Kornev. 2023. [ChatGPT vs. crowdsourcing vs. experts: annotating open-domain conversations with speech functions](#). In *Proceedings of the 24th Annual Meeting of the Special*

- Interest Group on Discourse and Dialogue*, pages 242–254.
- Nick Pangakis and Sam Wolken. 2025. [Keeping humans in the loop: human-centered automated annotation with generative AI](#). In *Proceedings of the Nineteenth International AAAI Conference on Web and Social Media*, pages 1471–1492.
- Ayesha Qamar, Jonathan Tong, and Ruihong Huang. 2025. [Do LLMs understand dialogues? a case study on dialogue acts](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26219–26237.
- Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. [Snorkel: Rapid training data creation with weak supervision](#). *Proceedings of the VLDB Endowment*, 11(3):269–282.
- Ido Roll, Vincent Aleven, Bruce M McLaren, and Kenneth R Koedinger. 2007. [Designing for metacognition—applying cognitive tutor principles to the tutoring of help seeking](#). *Metacognition and Learning*, 2:125–140.
- Johnny Saldaña. 2025. *The coding manual for qualitative researchers*, 5 edition. SAGE publications Ltd.
- Isabel Schorr, Lisa Bardach, Babette Bühler, and Enkelejda Kasneci. 2026. [Learning-to-learn in the age of generative AI: A scoping review and conceptual framework](#). *Computers and Education: Artificial Intelligence*, 10:100575.
- Hope Schroeder, Deb Roy, and Jad Kabbara. 2025. [Just put a human in the loop? investigating LLM-assisted annotation for subjective tasks](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25771–25795.
- David Williamson Shaffer, Wesley Collier, and Andrew R Ruis. 2016. [A tutorial on epistemic network analysis: Analyzing the structure of connections in cognitive, social, and interaction data](#). *Journal of Learning Analytics*, 3(3):9–45.
- Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. [The ICSI meeting recorder dialog act \(MRDA\) corpus](#). In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004*, pages 97–100.
- Aseem Srivastava, Ishan Pandey, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. [Response-act guided reinforced dialogue generation for mental health counseling](#). In *Proceedings of the ACM Web Conference 2023*, pages 1118–1129.
- Abhijit Suresh, Jennifer Jacobs, Charis Harty, Margaret Perkoff, James H Martin, and Tamara Sumner. 2022. [The TalkMoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4654–4662.
- Paul Thomas, Gabriella Kazai, Ryen White, and Nick Craswell. 2022. [The crowd is made of people: Observations from large-scale crowd labelling](#). In *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval*, pages 25–35.
- Xinru Wang, Hannah Kim, Sajjadur Rahman, Kushan Mitra, and Zhengjie Miao. 2024. [Human-LLM collaborative annotation through effective verification of LLM labels](#). In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–21.
- Philip H. Winne and Allyson F. Hadwin. 1998. [Studying as self-regulated learning](#). In *Metacognition in Educational Theory and Practice*, pages 277–304. Lawrence Erlbaum Associates Publishers.
- Houyu Wu, Haiyang Ni, Wenfu Luo, and Tenglong Wu. 2026. [Development and validation of the AI dependence scale for Chinese undergraduates and a preliminary exploration](#). *Frontiers in Psychology*, 16:1725393.
- Shunan Zhang, Xiangying Zhao, Tong Zhou, and Jang Hyun Kim. 2024. [Do you have AI dependency? the roles of academic self-efficacy, academic stress, and performance expectations on problematic AI usage behavior](#). *International Journal of Educational Technology in Higher Education*, 21:34.
- Barry J Zimmerman. 2002. [Becoming a self-regulated learner: An overview](#). *Theory Into Practice*, 41(2):64–70.

A Annotation Scheme Development

The following section delineates the two phases of developing our annotation scheme.

A.1 Initial Annotation Scheme Creation

The initial version of the coding scheme was strongly informed by the L2L dimensions defined in [Schorr et al. \(2026\)](#). In particular, the first author explored whether categories from the cognitive and metacognitive dimensions (either individually or by collapsing closely related subcategories into broader groups) could be applied to randomly selected chatbot interaction data.

However, it quickly became apparent that these constructs are not directly observable in chatbot interaction logs. Consequently, the first author simplified the coding scheme to focus on more reliably identifiable interactional behaviors. The resulting initial scheme consists of three main dimensions:

(1) relation to the assignment or expected deliverable, (2) interaction orientation (including a distinction between epistemic and performative interactions, capturing differences between knowledge-seeking and solution-seeking behavior), and (3) epistemic engagement continuity.

We designed the codes of the initial annotation scheme to capture behavioral indicators of cognitive and metacognitive engagement with the assignment, as well as the extent and consistency of students' interaction with the chatbot during task completion.

A.2 Annotation Scheme Refinement

During codebook refinement, two to three annotators independently labeled randomly selected subsets of the data, computed inter-rater agreement, and discussed disagreements to identify ambiguities and revise code definitions. One annotator left the project after the initial refinement rounds, resulting in two annotators for the final two iterations. This iterative process progressively transformed an initially independent label set into a more structured, decision-tree-like annotation scheme. The inter-rater agreement improved substantially over the course of five refinement rounds. Early iterations showed inconsistent agreement (Krippendorff's α between -0.02 and 0.67 across labels). After the introduction of the final codebook version, the last two refinement rounds yielded Cohen's κ values of 0.51 and 0.77 (Table 1), with the latter achieved without further changes to the scheme. Lower early agreement reflects ongoing alignment of the shared interpretation of the labels.

Table 5 summarizes the codebook's development trajectory across refinement phases, listing the codes, the observed reliability, the empirical reason for change, and the type of revision.

B Final Annotation Scheme

B.1 Annotation Codebook

Table 6 and 7 summarize the final operational definitions and decision rules used during annotation, including representative examples.

B.2 Theoretical Motivation of Annotation Scheme

We designed our annotation scheme to capture cognitively (in particular L2L) motivated aspects while remaining applicable to naturally occurring student-chatbot interactions. These labels are behavioral

proxies for cognitive and metacognitive processes rather than direct measurements of learners' internal states. The scheme properties and motivations are as follows:

- The labels are subject-agnostic but not task-agnostic: they can be applied across domains (e.g., mathematics, programming, writing), but the same utterance may receive different labels depending on the task context and expected solution space. This reflects the fact that learner intent is shaped by task structure rather than subject matter alone.
- The distinction between *solution seeking* and *knowledge seeking* seeks to capture forms of cognitive offloading in educational dialogue. However, the following needs to be considered: While solution-seeking behavior may indicate an intent to obtain a directly usable answer (compare Figure 2), it can overlap with knowledge acquisition, as solutions in educational settings often serve as a proxy for understanding. Because this distinction is not always recoverable from surface form alone, *solution seeking* is prioritized in the annotation decision process when ambiguity arises, reflecting its relevance for identifying potential over-reliance on external problem-solving.
- *Solution type* (create vs. verify/fix/adapt) is associated with different (meta)cognitive processes involved in learning. Creation typically involves planning, retrieval, and generation activities, whereas verify/fix/adapt involves evaluation, error detection, and refinement. In the context of learner-chatbot interactions, these requests correspond to different forms of help-seeking behavior and may map onto different pedagogical roles of the chatbot, such as tutor or scaffold versus editor or optimizer.
- *Cognitive engagement* captures whether learners actively interact with prior information and discussions or merely issue isolated requests. This dimension reflects whether the chatbot is used as a conversational partner for knowledge development or primarily as an oracle for information retrieval.
- The *mental model* dimension captures explicit or implicit reflection on existing knowledge, representing a key aspect of metacognitive awareness in L2L. It reflects whether learners

Phase	Scheme	Reliability	Reason for change	Revision
0	L2L cognitive/metacognitive dimensions (Schorr et al., 2026) – motivation only	-	Constructs not reliably observable in chatbot logs	-
1	8 independent codes, multi-label: epistemic engagement continuity + 7 engagement types (epistemic; performative: construction, refine/verify ($\times 3$ artifact origin types); metacognitive: planning, reflection)	Krippendorff’s α : -0.02 to 0.67 (3 annotators)	Highly uneven reliability, especially performative and metacognitive categories; difficulties distinguishing who produced an artifact; difficulties judging epistemic engagement for performative interactions	Change for Phase 2: Drop origin type; split epistemic engagement continuity
2	Consolidated to 7 codes, multi-label: 2 continuity dimensions (epistemic, performative) + 5 engagement types (epistemic; performative: construction, refine/verify; metacognitive: planning, reflection)	Krippendorff’s α : -0.01 to 0.48 (3 annotators)	Still unreliable, especially the performative continuity dimension and the metacognitive subtypes; epistemic and performative overlap	Change for Phase 4: Introduce hierarchical single-label structure; epistemic $\rightarrow$ knowledge seeking; performative $\rightarrow$ solution seeking; limit engagement and metacognition to knowledge seeking as a student property (cognitive engagement and mental model)
3	Reorganized around context dependence, learner intent (understanding vs. offloading), request mode (elicitation vs. evaluation)	Discarded before testing	Unreliable even after repeated first-author review; abandoned pre-testing	-
4	Hierarchical decision tree, single-label (final structure)	Cohen’s κ = 0.55 (2 annotators)	-	-
5	Hierarchical decision tree, single-label (final structure)	Cohen’s κ = 0.76 (2 annotators)	-	-

Table 5: Evolution of the annotation scheme across development phases. Reliability scores and revisions document the transition from independent multi-label annotation to the final hierarchical single-label structure.

externalize, test, or revise their current understanding during interaction.

Together, these labels reflect both cognitive processes (e.g., problem-solving, information seeking) and metacognitive behaviors (e.g., reflection, monitoring) in student-LLM interactions.

C Computational Details

GPT-5.4 was accessed through the OpenAI API. GPT-OSS:120B was run locally on a single NVIDIA H200 NVL GPU (141 GB VRAM), served via Ollama with a context length of 32,768 tokens and MXFP4 quantization applied to the MoE weights. For the 408 instances annotated by both LLMs and both human annotators, GPT-5.4 processed 2,427,870 input tokens and 119,241 output tokens over approximately 63 minutes, at an estimated cost of \$7,86 USD, calculated retrospec-

tively from the OpenAI API list pricing in effect at the time of use, as costs were not logged directly. Corresponding token and runtime figures for GPT-OSS:120B could not be reliably recovered and are therefore omitted; as it was run on local infrastructure, no direct monetary cost was incurred.

D DSPy Label Signatures

This section provides details on the DSPy signatures used for automated annotation.

D.1 Assignment

```
class AssignmentClassifier(dspy.Signature):
    """
    You are annotating a student-chatbot
    interaction from a university AI course.
    You are given multiple assignment
    descriptions and a student conversation.

    Each assignment is labeled (a1-a7). Compare
    the student's current prompt history to
```

Dimension	Category	Definition & Decision Rule	Example
Assignment Relation	Setting-up	Interaction addresses logistical, technical, or procedural conditions for task engagement (e.g., platform use, workflow coordination) rather than task content.	"ModuleNotFoundError: No module named 'tqdm'"
	Related	Interaction connects to task content, descriptions, expected prior knowledge, or learning objectives, incl. direct continuations of a prior task-relevant exchange.	"How to check if my given datasets have any duplicate rows? they are already loaded" (see assignment a3: <i>Part 3.1: Drop Duplicate rows</i>)
	Unrelated	Interaction does not substantively connect to task content or assignment-relevant knowledge.	"Can you remember data from previous chat"

Table 6: Compact codebook: category definitions, decision rules, and illustrative learner-chatbot examples for assignment relation.

```

the assignment descriptions.

Select the assignment that best matches the
student's inquiries. If no clear match
exists, return the original assignment.
"""
assignment_descriptions:dict = dspy.
    InputField(desc="Summaries of
        assignments")
chat:str = dspy.InputField(desc="Prompt
    history")
original_assignment:str = dspy.InputField(
    desc="Originally assigned assignment")

assignment:Literal["a1", "a2", "a3", "a4", "
    a5", "a6", "a7"] = dspy.OutputField()

```

D.2 Relation to the Assignment

```

class PromptRelation(dspy.Signature):
    """
    You are annotating a student-chatbot
    interaction from a university AI course.
    Classify how the student's current
    prompt relates to the given assignment.

    You are given:
    - assignment_context: description of the
        assignment, requirements, and expected
        outcomes
    - history: previous conversation between
        student and assistant
    - prompt: the student's latest message to
        classify

    Choose one label:

    - setting up the learning environment:
        The prompt is about setup, tooling,
        infrastructure, or logistics needed
        to work on the assignment (e.g.,
        installing Python, configuring IDEs,
        accessing datasets, debugging
        environment issues), even if it is in
        service of the assignment.

    - related:
        The prompt directly or indirectly relates
        to the assignment content,
        requirements, concepts, expected

```

```

deliverables, or builds on prior
assignment-related conversation.

- unrelated:
    The prompt has no meaningful connection
    to the assignment context, goals, or
    required knowledge, and is instead
    about unrelated topics or general
    conversation.
"""
assignment_context: str = dspy.InputField(
    desc="Assignment description,
        requirements, and objectives")
history: str = dspy.InputField(desc="Previous
    chat history")
prompt: str = dspy.InputField(desc="Current
    student message")

relation: Literal["related", "unrelated", "
    setting up the learning environment"] =
    dspy.OutputField(
        desc="Classify relation to assignment"
    )

```

D.3 Solution Seeking

```

class DecisionSolutionSeeking(dspy.Signature)
    :
    """
    You are annotating a student-chatbot
    interaction from a university AI course.
    Determine whether the learner's prompt
    is aimed at obtaining a concrete
    solution to an assignment task or
    subtask.

    This includes cases where the user:
    - asks for code, final answers, or
        implementations that directly produce a
        solution to an assignment task or
        subtask
    - requests help that would reasonably result
        in a full or partial solution (even if
        framed as a question for explanation,
        illustration, or examples)

    Do NOT classify as solution-seeking if the
    user is only:
    - asking conceptual questions without
        expecting an answer to be produced

```

Dimension	Category	Definition & Decision Rule	Example
Solution Seeking	Yes	Learner requests the chatbot to directly produce, verify, fix, adapt, or otherwise assist with obtaining a solution to an assignment (sub)task.	"# Export the dataframe to a new csv file dfoptimized.to_csv() # Export the dataframe to a new json file dfoptimized.to_json() what should be added to this"
	No	Learner does not request assistance in directly producing, verifying, fixing, or adapting a solution to an assignment (sub)task.	"is k fold cross validation already used in cross val score"
Solution Seeking Type (if Solution Seeking = Yes)	Create	Learner requests a <i>new</i> working solution for an assignment (sub)task, even if a code frame is already provided.	"# Export the dataframe to a new csv file # Export the dataframe to a new json file"
	Verify/Fix/Adapt	Learner asks to verify, fix, or adapt an <i>existing</i> solution (own or bot-provided), incl. via an error message or a follow-up modification request.	"# Export the dataframe to a new csv file csv_file_path = [...] im getting error with this"
Knowledge Seeking (if Solution Seeking = No)	Yes	Learner does not seek a solution but seeks information, explanation, or understanding.	"what are activation functions, alpha, and solvers in relation to neural nets" (general conceptual question, where the answer is not a solution to a specific assignment)
	No	Learner neither requests a solution nor seeks conceptual understanding or explanation.	"hi"
Mental Model (if Knowledge Seeking = Yes)	Yes	Learner proposes, tests, or refines their own assumptions, hypotheses, or interpretation.	"Does scores represent the score on the test data for each of the random splits"
	No	Learner requests information or explanation without referencing their own reasoning or assumptions.	"what does the scaler algorithm do"
Cognitive Engagement (if Knowledge Seeking = Yes)	Yes	Clear evidence the learner engaged with the content of the preceding bot message.	Bot writes "X.loc[:, [...]]"; learner asks "what is the :, doing in the x.loc" (a direct follow-up on the chatbot response)
	No	Learner moves the interaction forward without engaging with the prior response's content.	Unrelated question after a discussion about dataframe manipulation: "in python if I have a string like this 'hello_world' and I want to replace what every is before the '_' with another string how would I do it?"

Table 7: Compact codebook: category definitions, decision rules, and illustrative learner-chatbot examples for each annotation dimension of the main label.

- discussing theory without needing an executable or concrete output
- asking for output where the connection to assignment tasks is ambiguous

Do not infer solution-seeking based on potential downstream use of the response. Only label as solution-seeking when the prompt is clearly anchored in solving an assignment task or subtask itself, not when it merely could be used to construct a solution.

Use assignment_context and history to disambiguate intent.

```
assignment_context: str = dspy.InputField(
    desc="Task excerpts by similarity")
history: str = dspy.InputField(desc="Prior
conversation context")
prompt: str = dspy.InputField(desc="Current
student message")
```

```
solution_seeking: Literal["yes", "no"] = dspy
.OutputField(
```

```
desc="yes if the user is trying to obtain
a concrete solution"
)
```

D.4 Solution Type

```
class DecisionSolutionType(dspy.Signature):
    """
    You are annotating a student-chatbot
    interaction from a university AI course.
    The learner's prompt has been
    identified as solution seeking behavior.
    Classify the type of solution-related
    intent.
```

Choose:

- create:

The learner is requesting a new solution, implementation, or answer that has not yet been provided (even if a code frame is given by the assignment task).

- verify/fix/adapt:

The learner is working with an existing solution (from assignment or history)

```

        and wants to:
        - debug it
        - correct it
        - improve it
        - adapt it to new constraints
        - verify correctness
    """
    assignment_context: str = dspy.InputField(
        desc="Task excerpts by similarity")
    history: str = dspy.InputField(desc="Prior
    conversation context")
    prompt: str = dspy.InputField(desc="Current
    student message")

    solution_type: Literal["create", "verify/fix/
    adapt"] = dspy.OutputField(desc="create
    or verify/fix/adapt")

```

```

- no:
    The learner:
    - issues standalone requests
    - shows no connection to prior
      conversation
    - does not build on earlier content
    """
    history: str = dspy.InputField(desc="Previous
    conversation history")
    prompt: str = dspy.InputField(desc="Current
    student message")

    cognitive_engagement: Literal["yes", "no"] =
    dspy.OutputField(
        desc="yes if there is clear engagement
        with prior context"
    )

```

D.5 Knowledge Seeking

```

class DecisionKnowledgeSeeking(dspy.Signature
    ):
    """
    You are annotating a student-chatbot
    interaction from a university AI course.
    Determine whether the learner is
    seeking understanding, explanation, or
    information.

    - yes:
        The learner asks for information,
        explanations, definitions, concepts,
        reasoning, or clarification.

    - no:
        The learner's request cannot be
        reasonably interpreted as knowledge
        seeking.
    """
    prompt: str = dspy.InputField(desc="Current
    student message")

    knowledge_seeking: Literal["yes", "no"] =
    dspy.OutputField(
        desc="yes if the user is reasonably
        exhibiting knowledge seeking behavior
    ")

```

D.6 Cognitive Engagement

```

class DecisionCognitiveEngagement(dspy.
    Signature):
    """
    You are annotating a student-chatbot
    interaction from a university AI course.
    Detect whether the learner shows active
    cognitive engagement with prior context.

    - yes:
        The learner:
        - reacts to previous assistant messages
        - asks follow-up questions
        - challenges, refines, or builds on prior
          answers
        - references earlier steps or reasoning

```

D.7 Mental Model

```

class DecisionMentalModel(dspy.Signature):
    """
    You are annotating a student-chatbot
    interaction from a university AI course.
    Determine whether the learner is
    expressing or refining an internal
    understanding (mental model) of the
    subject. A mental model is present
    if the user is testing or proposing how
    a system works (even implicitly).

    - yes:
        The learner:
        - proposes explanations or hypotheses
        - expresses assumptions or reasoning
        - compares alternative ideas
        - tries to validate or refine their
          understanding
        - asks if their interpretation is correct

    - no:
        The learner:
        - only requests information or
          explanations
        - does not reference their own reasoning
          or assumptions
        - passively asks for answers without self
          -explanation
    """
    prompt: str = dspy.InputField(desc="Current
    student message")

    mental_model: Literal["yes", "no"] = dspy.
    OutputField(
        desc="yes if user expresses or refines an
        internal model of understanding"
    )

```

E Details on Prompting Setup

During prompting the LLMs for annotation, we incorporate the following inputs for each subtask:

- Assignment: summaries of all assignments, the full chat history, and the original assignment mapping

- Relation to the Assignment: summary of the relevant assignment, limited chat history, and the current student utterance
- Solution Seeking / Solution Type: top-5 most similar assignment chunks retrieved via embedding-based similarity, along with limited chat history and the current utterance
- Knowledge Seeking / Mental Model: current student utterance only
- Cognitive Engagement: limited chat history and current student utterance

This structured prompting strategy allows us to tailor contextual information to each decision step while maintaining a reasonable context size.

F Details on Retrieval Setup

Markdown (.md) files are segmented by headings and further sub-chunked into passages of at most 2000 characters while preserving word boundaries. Python (.py) files are parsed using abstract syntax tree (AST) structure to extract functions, methods, and remaining top-level code, as these often correspond to student-relevant implementation units. Jupyter notebooks (.ipynb) are split by cell, with markdown cells additionally sub-chunked using the same length constraint as markdown files. Other file types (e.g., .csv, .txt) are excluded from chunking as they primarily contain dataset artifacts, which we do not provide to the annotation models to limit the context length.

G Details on Chat History Construction

We prepare the chat history for each annotation decision as follows:

- Assignment: We use the full chat history, but retain only student prompts, while chatbot responses are replaced with "...". Since student prompts often include long copied assignment descriptions, we truncate each utterance to its first and last 150 characters.
- Relation to the Assignment: We include the preceding 5 prompt-response pairs. Only student prompts are kept, while chatbot responses are replaced with "...". Utterances are truncated to 250 characters at both ends. Additionally, prior predictions for *assignment relation*, *solution seeking*, and *solution type* are appended to the history.

- Solution Seeking / Solution Type: We again use the preceding 5 prompt-response pairs. Chatbot responses are replaced with "...", except for the most recent response to provide context in case the student references it. Student utterances are truncated to 250 characters at both ends. Previous predictions for *assignment relation*, *solution seeking*, and *solution type* are included.
- Cognitive Engagement: We use the preceding 5 prompt-response pairs, replacing all chatbot responses with "...", except for the immediate prior response again, as it might be necessary for reference. Student utterances are again truncated to 250 characters at both ends.

H Label Distribution and Per-Class Performance

Before comparing agreement scores, we first examine the distribution of the main labels in the double-coded human-annotated data. This is important because both accuracy and Cohen’s κ are affected by class imbalance, particularly in hierarchical annotation schemes where some decision paths occur substantially more often than others. Table 8 reports the support of each main label in the consolidated human annotations.

To complement aggregate agreement scores, we also report per-class precision, recall, and F1-score for the two LLM annotators against the consolidated human labels. These metrics reveal whether model performance is concentrated on frequent labels or remains stable across less frequent learner-intent categories. Similar to Cohen’s kappa results, we observe weaker agreement on infrequent labels (Table 9). This effect is especially pronounced for classes involving a final "yes" decision in the *mental model* path, where agreement deteriorates.

I Disagreement Details

This section details the reasons for disagreements between model (GPT-5.4) and human annotations by identifying the factors that led the model to reach a different conclusion.

I.1 Model Limitations

The error patterns attributable to model limitations identified in our observed data span five categories: context ignored, reasoning failure, implicit intent missed, hallucinations, and task relation missed.

Main label	Support	%
Solution seeking: create	147	36.03
Solution seeking: verify/fix/adapt	140	34.31
Knowledge seeking + cognitive engagement + no mental model	45	11.03
Knowledge seeking + no cognitive engagement + no mental model	39	9.56
Knowledge seeking + cognitive engagement + mental model	13	3.19
Knowledge seeking + no cognitive engagement + mental model	1	0.25
Other / non-knowledge seeking	23	5.64
Total	408	100.0

Table 8: Distribution of consolidated human main labels in the double-coded evaluation subset.

Main label	P _{5.4}	R _{5.4}	F1 _{5.4}	P _{120B}	R _{120B}	F1 _{120B}
Solution seeking: create	0.87	0.71	0.78	0.66	0.80	0.72
Solution seeking: verify/fix/adapt	0.67	0.74	0.70	0.83	0.41	0.55
Knowledge seeking + cognitive engagement + no mental model	0.52	0.62	0.57	0.37	0.60	0.46
Knowledge seeking + no cognitive engagement + no mental model	0.56	0.38	0.45	0.45	0.26	0.33
Knowledge seeking + cognitive engagement + mental model	0.28	0.54	0.37	0.17	0.38	0.24
Knowledge seeking + no cognitive engagement + mental model	0.00	0.00	0.00	0.00	0.00	0.00
Other / non-knowledge seeking	0.18	0.22	0.20	0.28	0.43	0.34
Macro average	0.44	0.46	0.44	0.39	0.41	0.38
Weighted average	0.67	0.64	0.65	0.63	0.56	0.56

Table 9: Per-class precision, recall, and F1-score for main-label prediction by both LLM annotators, evaluated against consolidated human labels.

The model predictions diverge from the human annotations due to the following reasons:

- **Context ignored.** The model occasionally fails to incorporate relevant contextual information, particularly task excerpts in *solution type* detection, even when such information is correctly used in *solution seeking* detection. This issue is most prominent in assignments involving open-ended analysis or argumentative deliverables.
- **Reasoning failure.** In some cases, the model produces decisions that lack coherent justification or contradict explicit cues in the input. For instance, non-solution-oriented formatting instructions are misclassified as *knowledge seeking*, or explicit non-solution-oriented prompts (e.g., "just read this") are incorrectly labeled as *solution seeking*.
- **Implicit intent missed.** The model tends to over-rely on surface-level phrasing, failing to infer implicit learner intent. This includes cases where task descriptions are phrased as questions or where prompts are copied verbatim from assignments, leading to misclassification as *knowledge seeking* despite underlying solution-oriented intent.
- **Hallucinations.** The model introduces information not present in the dialogue context,

attributing decisions to nonexistent prior references. We interpret these cases as grounding failures, suggesting that future work could improve grounding by requiring the model to cite supporting evidence spans or abstain when no supporting context exists. These potential mitigations were not evaluated in this study.

- **Task relation missed.** The model overgeneralizes topical similarity, treating prompts as assignment-related even when they fall outside the actual course scope.

Table 10 illustrates each case with a representative example.

I.2 Task and Annotation Design Issues

In some cases, neither the model nor the human annotators are at fault. Instead, the issue stems from the annotation pipeline design, such as label definitions or insufficient contextual information. We categorize these cases as follows:

- **Label misinterpretation.** The model exhibits deviations in how label definitions are operationalized. For example, it may interpret *solution seeking* narrowly as requesting entirely new solutions, excluding reformulation or refinement requests of existing outputs (e.g., re-formatting, rewriting, or adapting analyses). Similarly, it inconsistently applies distinctions

between *solution types* (i.e., *create* vs. *verify/fix/adapt*), particularly when extending partial solutions. For *cognitive engagement*, the model inconsistently assesses relatedness to prior context, at times overgeneralizing weak topical connections while in other cases failing to recognize explicit links.

- **Missing context.** Errors arise from insufficient or misaligned contextual information. For instance, relevant assignment excerpts may not be collected during retrieval when prompts are underspecified (e.g., "do the same for this method") or dominated by large code blocks, leading to incorrect *solution seeking* classifications. Additionally, the required context may be unavailable altogether, as we do not provide the model with data file contents (.csv, .txt). Conversely, the absence of dialogue history can obscure knowledge-seeking intent in follow-up or clarification utterances. Similar issues occur when assignment examples are misinterpreted as student-generated content during *cognitive engagement* decisions due to the model not having access to assignment content.
- **Prompting misread.** In some cases, the model misinterprets the annotation prompt itself, treating meta-instructions (e.g., output format requirements) as part of the learner utterance.

Table 11 illustrates each case with a representative example.

I.3 Data Ambiguity and Human Factors

Lastly, disagreement may arise from the inherent interpretive nature of the labels or data, or from human annotation errors, rather than from limitations in model or task design. We describe all three cases below:

- **Interpretive label ambiguity.** Some instances admit multiple valid interpretations, leading to divergent but defensible labels. This cause of disagreement is evident in *solution seeking*, where implicit or preparatory requests (e.g., providing context before an explicit task, requesting test cases, or task-enabling conceptual questions) may or may not be considered solution-oriented. Similar

ambiguities arise in *solution type* (e.g., formatting as refinement vs. explicit task requirement), *knowledge seeking* (e.g., meta-questions about system capabilities or procedural tasks such as mathematical conversions), *cognitive engagement* (i.e., deciding whether a learner's contribution still belongs to the same cognitive instance or constitutes a topic shift), and *mental model* (e.g., whether brief factual statements may or may not reflect underlying understanding). In these cases, both human and model predictions can be plausibly justified.

- **Unclear student intent.** Ambiguity in learner intent further contributes to disagreement. This issue includes cases where students copy-paste assignment content without explicit requests, provide partially completed tasks without clear goals, or pose knowledge-oriented questions that may implicitly support solution development. Such instances make it difficult to distinguish between background provision, self-directed problem-solving, and implicit requests for solutions.
- **Human labeling error.** Variability and errors in human annotation contribute to disagreement, including reasoning errors, overlooked contextual relations, and shifts in annotation criteria over time. This issue is particularly evident for more interpretive labels, such as *cognitive engagement* and *mental model*, where annotator understanding evolved during the annotation process.

Table 12 illustrates each case with a representative example.

J Disagreement Distributions

We provide detailed breakdowns of disagreement categories across *solution seeking* (Figures 4 and 5), *solution type* (Figure 6), and *knowledge seeking* sub-decisions (Figure 7), illustrating how different sources of error (model behavior, human annotation, and data ambiguity) contribute under each aspect.

For *solution seeking* disagreements where the human labels "true" but the model predicts "false," errors are mainly due to model misinterpretation of labels or student intent, as well as ambiguous expressions of intent. Conversely, when the model predicts "true" but the human labels "false," disagree-

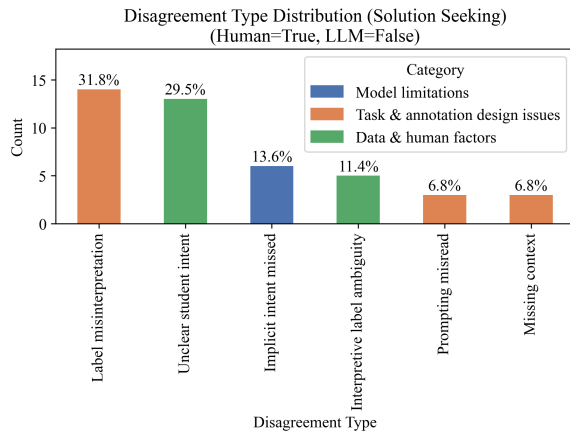


Figure 4: Distribution of disagreement reasons for cases where human annotators predicted *solution seeking* but GPT-5.4 did not (percentages indicate share of assigned labels, not instances). Disagreements are mainly due to model misinterpretation of labels or student intent, as well as ambiguous expressions of intent.

ments are largely driven by inherent data and label ambiguity, where both interpretations can be considered valid. In *solution type* disagreements, errors are primarily attributable to prompting and model behavior, alongside unavoidable label ambiguities that cannot be resolved through improved modeling or annotation design. In contrast, *knowledge seeking* disagreement cases are predominantly explained by human annotation errors, where model predictions remain largely acceptable.

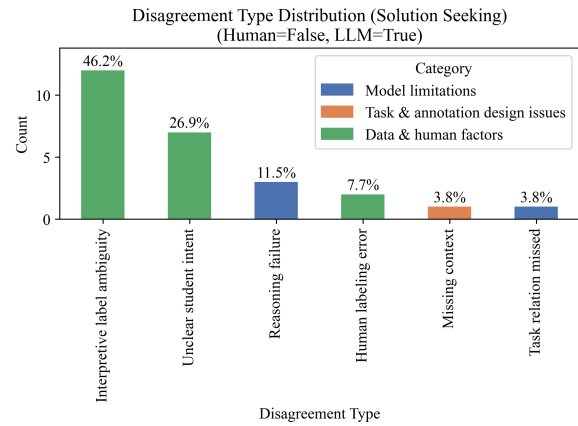


Figure 5: Distribution of disagreement reasons for cases where GPT-5.4 predicted *solution seeking* but human annotators did not (percentages indicate share of assigned labels, not instances). Disagreements are largely driven by inherent data and label ambiguities, where both interpretations may be considered valid.

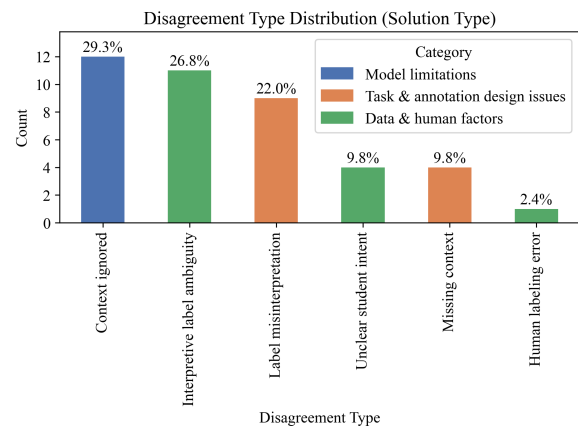


Figure 6: Distribution of disagreement reasons for *solution type* (percentages indicate share of assigned labels, not instances). Disagreements are largely driven by model errors or ambiguous cases rather than human annotation issues.

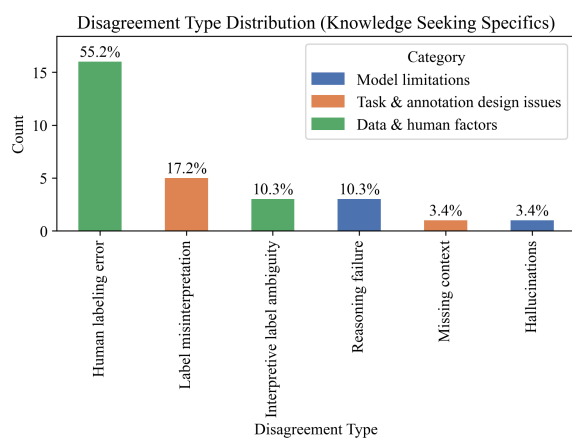


Figure 7: Distribution of disagreement reasons for *knowledge seeking* subcategories (percentages indicate share of assigned labels, not instances). Human labeling errors dominate, while task design appears largely adequate, and model performance is comparatively stable.

Disagreement Type	User Request	LLM Reasoning (GPT-5.4)	Assessment
Context ignored	"Explain your intuition in recursive DFS VS stack-based DFS, and which one you used here. My code is below: (pasted code)"	Verify/fix/adapt: The student is asking for help explaining the difference between recursive DFS and stack-based DFS in relation to their existing 'suggest_dfs' implementation. They provided code and want explanation/verification of which approach they used, not a brand-new solution. This is adapting/verifying an existing solution and helping describe it for the assignment writeup.	The model treats the request as a general explanation of existing code for verification but overlooks that the assignment explicitly requires this explanation as part of completing a task. The context given to GPT-5.4 indicates that the student is requesting help with solving an assignment deliverable from scratch.
Implicit intent missed	"How to normalized numerical features in pandas"	No solution seeking: The user asks a conceptual/how-to question about normalizing numerical features in pandas. Although the assignment context includes a TODO about normalizing features, the prompt does not ask for the specific code to complete that assignment cell or for a concrete solution on the provided dataset. It is general guidance rather than a direct request for an assignment answer.	The LLM recognizes the relationship to an assignment task but relies on the surface-level phrasing instead of inferring the student's likely intent to complete it.
Hallucinations	"nn3 = MLPClassifier([...] hidden_layer_sizes=(10, 2), [...]) [...] what does (10,2) represent"	Cognitive Engagement: The learner asks a follow-up question about code shown in the ongoing neural network discussion, specifically referencing the parameter 'hidden_layer_sizes=(10, 2)' from prior context. This builds directly on the earlier exchange about neural network configurations and performance, so it shows active engagement with prior context.	The cited "hidden_layer_sizes=(10, 2)" does not appear in the preceding chatbot response, so the LLM incorrectly assumes a prior reference that is not present.
Task relation missed	"[...] TypeError: 'ChatCompletionMessage' object is not subscriptable"	Solution seeking: The student is asking about a concrete coding error encountered while implementing API calls over a dataframe in the context of the assignment's GPT data manipulation task. Fixing the TypeError would directly enable their assignment code to run and produce outputs needed for the task, so this is clearly help toward a concrete subsolution rather than a purely conceptual question.	The requested deliverable is outside the scope of the course and therefore cannot constitute <i>solution seeking</i> . The LLM incorrectly infers assignment relevance.
Reasoning failure	"Just read this (pasted assignment task implementation)"	Solution seeking: The student provides a concrete implementation for the assignment's "Section 3.5: Hyperparameter Tuning," including training code, evaluation code, five experiment settings, and result tabulation. This is directly tied to completing the assignment subtask rather than asking a conceptual question. The prompt is therefore clearly aimed at obtaining or presenting a concrete solution to the assignment component.	The student explicitly asks the model only to read the provided content and does not request any solution or assistance with the assignment. The LLM incorrectly infers <i>solution seeking</i> from the assignment-related code rather than reasoning over the explicit "just read this" request.

Table 10: Illustrative examples for the disagreement types: model limitations.

Disagreement Type	User Request	LLM Reasoning (GPT-5.4)	Assessment
Label misinterpretation	"first person"	No solution seeking: The user asks for a stylistic rewrite ("first person") of an explanation for code already discussed in the assignment context. This is still directly tied to presenting/justifying their assignment solution, but it does not request new code or a new concrete solution. Since the request is only to rephrase explanation text rather than solve a task/subtask, it is not solution-seeking.	Because the explanation forms part of the assignment response, this request qualifies as <i>solution seeking</i> . The category is not meant to be limited to requests for a new solution. However, the definition in the labeling prompt is ambiguous enough to support the interpretation the LLM adopted.
Missing context	"(pasted code implementation for an assignment task) give the code analysis"	Verify/fix/adapt: The student provided an existing implementation of 'calculate_probability' and asked for "the code analysis," which means they want explanation/review of code they already have rather than a brand-new solution. This is analysis/verification of an existing solution.	While the assignment excerpts explain the function's purpose and parameters, the relevant assignment context for the requested code analysis is missing from the provided excerpts.
Prompting misread	"(pasted assignment information)"	No solution seeking: The user is not asking for code, an implementation, or a concrete answer to an assignment subtask. They pasted assignment background and then explicitly asked for the annotation-format output. So the intent is to classify the interaction, not to solve the autocomplete assignment itself.	The LLM misidentified the text to annotate, treating the annotation instructions as part of the student request.

Table 11: Illustrative examples for the disagreement types: task and annotation design issues.

Disagreement Type	User Request	LLM Reasoning (GPT-5.4)	Assessment
Interpretive label ambiguity	[student pasted partially completed code for an assignment step]: "[...] score = reg.score() what parameters should I use for score? X_train?"	Verify/fix/adapt: The student already has an existing code attempt and is asking how to correct/adapt it, specifically what arguments to pass to 'reg.score()'. This is a request to fix/verify usage of an existing solution rather than generate a brand new one.	The student provides existing code, but the missing arguments could be viewed either as a correction to an existing solution or as a missing solution component that needs to be created.
Unclear student intent	[student pasted an incomplete assignment template with one informal comment ("That part should be pretty easy tho.") and did not explicitly ask for anything, such as solutions for the unfinished report, analysis, or output sections]	No solution seeking: The user provides the assignment report template and a filled-in tree diagram, then says "That part should be pretty easy tho." They are not explicitly asking for code, answers, or writeups to complete any assignment subtask. This looks like sharing context rather than requesting a concrete solution. Since there is no direct request for producing report text, code analysis, outputs, or other assignment content in this prompt, it is not clearly solution-seeking.	The copied assignment does not contain an explicit request. It could be context for a later question or an indirect request for a solution, making the student's intent ambiguous.
Human labeling error	[ongoing discussion on broadcasting in Torch]: "<new example> how does this work??? 2 != 3, yes there's a 1, but that doesnt mean it should work???"	Cognitive engagement: The learner is clearly engaging with prior context. They reference the earlier broadcasting discussion, explicitly compare dimensions ("2 != 3, yes there's a 1"), and ask a follow-up question that challenges/refines their understanding of the previous explanation. This is not a standalone request; it builds directly on the earlier tensor-shape reasoning.	The LLM correctly identifies the student's <i>cognitive engagement</i> with the ongoing discussion on broadcasting.

Table 12: Illustrative examples for the disagreement types: data ambiguity and human factors.