Combining Dialog Acts and Skill Modeling: What Chat Interactions Enhance Learning Rates During AI-Supported Peer Tutoring?

Conrad Borchers
Carnegie Mellon University
cborcher@cs.cmu.edu

Kexin Yang
Carnegie Mellon University
kexiny@cs.cmu.edu

Jionghao Lin
Carnegie Mellon University
jionghal@cs.cmu.edu

Nikol Rummel
Ruhr-Universität Bochum
nikol.rummel@rub.de

Kenneth R. Koedinger
Carnegie Mellon University
koedinger@cmu.edu

Vincent Aleven
Carnegie Mellon University
aleven@cs.cmu.edu

ABSTRACT

Peer tutoring can improve learning by prompting learners to reflect. To assess whether peer interactions are conducive to learning and provide peer tutoring support accordingly, what tutorial dialog types relate to student learning most? Advancements in collaborative learning analytics allow for merging machine learning-based dialog act classification with cognitive modeling of fine-grained learning processes during problem-solving to illuminate this question. We estimate how much peer-tutored students improve in a collaborative tutoring system for linear equation-solving in K-12 mathematics in relationship to the peer dialog types they engage in. This work establishes a reliable BERT classifier with an accuracy of close to 80% to classify chat messages during peer tutoring into minimal, facilitative, and constructive, serving as instructional factors. Based on data from 394 students, peer tutor dialog was rare. Only 8% of tutee problem-solving steps were followed by peer tutor chat messages. Still, facilitative tutor dialog was associated with an increased tutee learning rate. Meanwhile, tutor dialog classified as constructive was associated with lower learning rates. Content analysis suggested that such dialog often reinforced incorrect solutions, gave away answers, or was unrelated to the taught content. Hence, considering problem-solving solution contexts could improve the assessment of peer tutoring dialog. Peer tutors engaging in little dialog could be attributed to the high cognitive demand of learning to tutor while still learning the content they tutor on. Providing peer tutors with instructional support to engage in constructive dialog may improve the tutee's learning.

Keywords

intelligent tutoring systems, peer tutoring, instructional factors analysis, cognitive modeling, dialog acts, collaborative learning analytics, natural language processing

C. Borchers, K. Yang, J. Lin, N. Rummel, K. R. Koedinger, and V. Aleven. Combining dialog acts and skill modeling: What chat interactions enhance learning rates during ai-supported peer tutoring? In B. Paaßen and C. D. Epp, editors, *Proceedings of the 17th International Conference on Educational Data Mining*, pages 117–130, Atlanta, Georgia, USA, July 2024. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.12729784

1. INTRODUCTION

Collaborative learning analytics is an emerging subdiscipline that leverages synergies with the long-standing research field of computer-supported collaborative learning (CSCL) [69]. Adopting learning analytics into CSCL goes hand-in-hand with using insights generated from collaborative learning processes to inform the redesign and adaptivity of collaborative learning systems. This desire to improve the educational impact of collaborative learning has sparked multiple successful lines of research. Among many, they include analytics to support effective group regulation [42], collaboration skills [21], and prompting for participation [57]. As the key application area of the present study, previous works investigated how data-driven artificial intelligence (AI) systems can support learners in peer-tutoring by offering adaptive support and improving tutor strategies [62, 63, 47, 49]. A promising way to achieve the potential of AI systems is by analyzing the chat behaviors within these peer tutoring sessions. This analysis is crucial for uncovering the nuanced ways in which collaborative learning occurs and evolves in CSCL environments [69].

Previous studies utilize dialog acts representing the intentions carried out through language to analyze chat behaviors. The analysis of dialog acts is crucial to understanding collaborative learning, particularly when they offer or elicit opportunities to learn [5]. Such opportunities arise when dialog allows students to reflect, explain, and challenge their position, exemplified by the concept of "accountable talk" from extensive classroom research by Resnick et al. [52]. Therefore, understanding when peer tutoring is effective likely lies in analyses of characteristics of dialog acts during peer tutoring conversations. Past research analyzing peer-tutoring processes offers ample support for this idea, documenting improvements in learning from positive encouragement [11] and sequences of positive feedback [61]. Similarly, past work has analyzed dialog during peer tutoring and delineated effective and ineffective collaboration dynamics to develop support systems for peer tutors, which have been shown to improve their learning significantly [63]. However, these prior works on effective dialog acts and processes primarily hand-coded dialog acts [11, 61, 63, 55] with the resulting sample sizes being too small to model learning [14]. Yet, the systems that support adaptive problem-solving during peer tutoring (e.g., intelligent tutoring systems; ITS) can

record learning processes on a fine-grained level. Modeling learning from these records in conjunction with dialog acts could illuminate the types of dialog acts learners benefit from most, the relative importance of dialog acts by tutors and tutees, and where collaborating students need more system-level support. In short, problem-solving log data could offer a quantified lens into the conditions under which peer tutoring is most effective and advance the scientific understanding of when collaborative learning yields learning conditions that are more favorable and effective than individualized problem-solving [31].

The present study merges two methodologies required to combine dialog acts with learner modeling: (a) annotations of dialog acts at larger scales and (b) their connection with fine-grained problem-solving log data annotated with underlying skill models. First, we discuss annotating dialog acts at scale. With natural language processing (NLP) advancements, automated classification of collaboration characteristics from text is feasible [54, 17]. Emerging work has also mined effective dialog acts from tutorial dialog represented as codes [36]. For example, in Vail et al. [61], an utterance "*No, it's wrong!*" from a tutor can be annotated as the dialog act `Negative Feedback`. Nascent work also demonstrates that automatically generated annotations for dialog acts are effective for understanding learning. Recent studies in learning analytics employed NLP techniques to analyze collaborative problem-solving, such as identifying collaborative skills through student speech [51], detecting language patterns in pair programming [60], and classifying interactions in collaborative science tasks [21].

Second, we discuss combining dialog acts with cognitive models of learning. One potential reason why instructional factors in collaborative learning have not been linked to cognitive modeling of learning is that most problem-solving environments do not produce problem-solving logs with steps annotated with underlying skill models [42, 13, 21, 57]. However, outside of collaborative problem-solving, learner models have a long-standing research track record in improving learning processes. Process log data of learning, specifically in ITS, allow for modeling learning rates (i.e., improvement rates on different skills relevant to a domain). Quantifying rates of learning have been routinely done with log data in ITS [15, 8, 31, 24] using an underlying skill model of cognitive operations associated with specific problem-solving steps. Learning is then measured by observing how likely a student is to correctly perform a problem-solving step requiring a given skill without needing system support (e.g., hints or feedback) [30]. The more often students encounter different problem-solving steps involving a given skill, the more likely they are to learn this skill. Instructional factors analysis (IFA) is an analytic method that can compare rates of acquisition as a function of *learning opportunities* related to different modes of instruction from which students learn when receiving feedback on their step attempts [14]. IFA has been used to understand learning in an online logical fallacy tutoring system comparing different types of explanation [15], model learning from reading in MOOCs [58], and quantify the effectiveness of collaborative problem-solving in elementary school mathematics [47]. In peer tutoring, IFA can be used to compare how much students improve when using or being exposed to different dialog acts and strategies.

The present study leverages advancements in classifying dialog acts from human-to-human tutorial dialog to account for differences in learning rates during problem-solving practice of linear equations in middle school peer tutoring. Prior work creating classifiers for dialog acts from text has used data sets from higher education (e.g., [17, 54]) or commercial tutoring contexts [55, 36]. Therefore, it is an open question whether peer tutoring chat messages from middle school students, which are expected to be shorter and less rich in linguistic features, can be reliably classified similarly. Automated classification of text artifacts from collaboration in combination with learner modeling is powerful in understanding why and when different forms of peer tutoring are most effective for learners. Yet, to the best of our knowledge, such a marriage between the classification of text and cognitive modeling has not been done. The present study considers different types of chat messages during peer tutoring as instructional events. We distinguish between minimal, facilitative, and constructive chats and whether the peer-tutor ("Tutor") or tutored student ("Solver") sends them. The present study's coding scheme was adapted from a scheme by Mawasi et al. [44], who identified classes of peer assistance behaviors in middle school mathematics. Following prior taxonomies in the student population featured in this study maximizes the probability of observing relevant variation in dialog acts of collaborating students and their relationship to learning rate differences of tutored students. We investigate how much students improve per learning opportunity (in terms of learning rates) when engaging in peer dialog compared to working alone. Such an intermediate comparison between paired and unpaired problem-solving establishes how much students learn between both conditions before diving into learning rate differences across specific instructional factors. We then ask how different classes of predicted chat messages in a collaborative tutoring system for linear equation-solving explain these differences in learning rates. Our three research questions are as follows:

RQ1: How accurately can students' chat messages in a collaborative tutoring system be classified into minimal, facilitative, and constructive using a BERT classifier?

RQ2: How do students' learning rates during problem-solving compare when paired with a peer tutor versus working individually with tutoring software?

RQ3: How does the tutored student's learning rate vary with the types of chat messages that the students exchange (i.e., minimal, facilitative, or constructive)?

2. RELATED WORK

The present study connects emerging methods for leveraging natural language in educational data mining with rich prior research studying collaborative learning by problem-solving in intelligent systems. Situated in the context of tutoring systems for collaborative learning, we additionally highlight past research on dialog acts in the context of peer tutoring that is relevant to CSCL.

2.1 Collaborative tutoring systems

Engaging in AI-supported dialog systems can help middle school students learn mathematics more effectively than working alone [63]. Student-to-student or peer tutoring involves

one learner solving problems while the peer tutor deepens their knowledge by revising the material they tutor on [62]. Prior work has built applications integrating effective peer tutoring principles, such as scaffolding writing for progress reports in group-based class assignments [64]. These environments can provide log data to study different forms of collaboration and their effectiveness via experiments [12]. The combination of theory-driven applications and their rich log data make peer tutoring suitable for studying dialog support during problem-solving.

Yet, one key challenge in analyzing text from dialog-based tutoring systems is establishing a coding scheme suited for a specific domain and student population. Past approaches for coding collaborative interactions ranged from focusing on facilitation aspects to categorizing types of participant interactions (for an overview, see [21]). The present study adopts a coding scheme from Mawasi et al. [44], who coded peer help-giving behaviors in middle school math classrooms with three categories for student conversation: *minimal*, *facilitative*, and *constructive contribution*. Help-giving is an integral part of many collaborative activities and is a critical element of the productive interactions identified by [26] that contribute to learning from collaboration.

Dialog-based tutoring systems can also feature intelligent agents that mimic human dialog. Latham et al. [34] developed an agent that uses dialog to adapt instruction to learners by detecting patterns in their learning behavior. Similarly, SimStudent is a teachable peer learner that allows a student to learn by teaching via chat and by giving the agent practice problems and test assessments [43]. Despite these advances, a literature review on pedagogical conversational agents [23] concludes that a proper generalization of existing design knowledge has not been synthesized from this line of research. Therefore, the present study derives design implications for an intelligent collaborative tutoring system by combining cognitive modeling of learning rates with students' use of chats during tutoring.

2.2 Dialog acts in tutoring

Identifying dialog acts (i.e., intentions carried out through language) from peer tutoring helps investigate student learning and evaluate tutoring quality [2]. The current study adopts a coding scheme from Mawasi et al. [44]. The three categories (i.e., minimal, facilitative, constructive contribution) are considered dialog acts in the current study. Using dialog acts to analyze tutoring dialog has been employed in many previous works [55, 10, 36] for revealing the effectiveness of dialog tutoring and investigating student learning performance. These works developed a coding scheme to annotate dialog acts by tutors and students manually. Prior research delineated effective dialog acts by analyzing the dialog acts from tutoring dialogs. For example, Boyer et al. [11] demonstrated that the dialog acts offering encouragement helped improve student problem-solving performance. Vail et al. [61] found that a sequential pattern of dialog acts, consisting of positive feedback given by tutors after a confirmation question from the student, was positively correlated with the learning gain of students. However, manually annotating dialog for many tutoring utterances is time-consuming and cost-demanding [37]. To address this issue, many empirical studies [55, 56, 36, 37, 38] annotated a certain amount of tutoring utterances and then employed supervised machine learning models to automate the annotating process. For example, Rus et al. [55] employed a conditional random field model to train on 500 Algebra tutorial sessions. Lin et al. [36] employed a BERT model trained on 45 tutoring sessions on mathematics topics. Inspired by the promising results of automating the identification of dialog acts, the present study aims to employ machine learning models to analyze the fine-grained differences (e.g., student learning rate) in learning, which is under-explored in the middle school peer tutoring context, as most past classifiers were trained on data sets from higher education (e.g., [17, 54]) or commercial tutoring contexts [55, 36].

2.3 Educational data mining for dialog

Several studies have leveraged rich log data from systems for collaborative problem-solving and intelligent tutoring to support and understand learning [29, 47, 49]. A common thread across both systems is that tutoring is realized through adaptive natural language, which differs from traditional tutoring systems that use hints and error feedback tailored to specific problem-solving steps. However, unlike the present study, these studies have not married cognitive modeling with rich linguistic data from interactions with these systems. Rather, a systematic literature review on ITS with natural language dialog [50] shows that most past studies used pre-to-post learning gains as measures, comparing learning via AB testing or controlled experiments, usually comparing different designs of an ITS. For example, prior research on EER-tutor [68] demonstrated how adaptive dialog that considers prior student errors can significantly improve learning gains. Similarly, research on Gaze Tutor [16] leveraged gaze data to detect and react to student disengagement by adapting dialog to re-engage the student. Nascent work by Abdelshiheed et al. [1] leveraged talk moves to predict assessment scores alongside ITS performance metrics.

Past work leveraged modeling approaches to support collaborative problem-solving. For example, Earle-Randell et al. [18] used Hidden Markov models to understand collaborative states of elementary school children learning in a collaborative block-based learning environment, finding that states of confusion or impasses most commonly related to exiting a state of productive talk. Multimodal learning analytics can also make collaborative learning more effective by detecting when team members have differing or insufficient opinions regarding task progress [41]. Yet, none of the surveyed literature has combined these emerging modeling approaches with students' moment-by-moment performance differences. The present study bridges this gap by combining traditional approaches to learner modeling in ITS [14, 31] with different types of tutoring strategies expressed in chat messages.

3. METHOD

Our methodology involves a multi-stage process whereby we create a classifier of peer-tutoring chat messages encoded as log data in an intelligent tutoring system for linear equation-solving. Generalizing that classifier then allows for fine-grain cognitive modeling of learning in relationship to the use and exposure to different types of chat messages sent by the Tutor and Solver. These chat types then serve as instructional factors in growth models of learning.

3.1 Study context

3.1.1 Sample

The study sample included data from two classroom studies in 2021 and 2023 in math classrooms in three public suburban middle schools in a mid-sized city in the east of the USA. A total of 394 students ranging from grades 6-8 and 10 teachers participated in the studies, totaling 22 classroom sessions. The ten teachers consist of seven female and three male teachers. The data collection aimed to evaluate the feasibility and desirability of dynamically combining individual and collaborative learning in math classrooms.

The research team recruited the schools through prior connections with local teachers, with study approval obtained from the administration of participating schools. The research followed an approved IRB protocol that fell under the exempt category of established educational settings and normal educational practices, hence requiring no consent. Teachers helped distribute letters to all caregivers ahead of the study, informing them about the data collection and giving them the opportunity to request their child's anonymized data to be removed from the research data set. In line with the approved IRB protocol, de-identified data is available upon request for research purposes through DataShop [30] by pooling three data sets mapping to the three study sites.[1]

3.1.2 Intelligent tutoring system for peer tutoring

APTA is a collaborative tutoring system for middle school equation-solving [63]. Students are paired in problem-solvers ("Solvers") and Tutors. The tutoring software provides immediate feedback and on-demand hints to assist the student in the Tutor role – it rarely interacts with the student in the Solver role. The Tutor is required to provide correctness feedback through the system interface (marking each step as correct or not) and can provide hints or explanatory messages via chat. The system intervenes when the Tutor gives incorrect correctness feedback, and a problem can only be finished once the Tutor has graded all steps. The student taking the Tutor role can request hints from the system for the Solver at any given step. Further, the system provides Tutors and Solvers with adaptive conversational support to support help-seeking and tutoring [63] This conversational support appears as messages sent by the computer in the chat panel of the Solver or Tutor (see Figure 1). The Solver and the Tutor do not necessarily sit close to each other. They can communicate fully online via a chat message box, the messages of which are analyzed in the present study. In the current design, students can only see their partners' alias usernames (e.g., "redlion") rather than real names. APTA has a track record of effectively supporting student learning [62, 63], but we note that the present study used a revised version of APTA, called APTA 2.0 [19]. A screenshot of APTA's interface is in Figure 1.

During individualized problem-solving, students worked with the Lynnette ITS for equation-solving [40]. Similar to APTA, Lynnette has an underlying knowledge component model that credits different skills corresponding to problem-solving step attempts of the Solver, which is used for cognitive modeling of learning (see Section 3.4). Specifically, the correctness of each problem-solving step attempt is related to one or more skills in Lynnette's skill model (e.g., "distribute-division"). Solvers can enter equations into Lynnette that compound multiple steps (e.g., directly entering the solution) and are credited for all steps bundled in a compound step, including all the skills associated with the bundled steps. For the present study, we used Lynnette's default skill model, whose empirical evaluation, including an open-source data set, is published in [40].

3.1.3 Procedures

Throughout all classroom sessions, lasting typically 50 minutes, students practiced linear equations while their teacher walked around the room to support students when needed. Researchers were present during the study to resolve technical issues. At the beginning of classroom sessions, all students started out practicing individually until the teacher started pairing students. Specifically, the teacher monitored and paired students during the class session through a desktop computer or a tablet. Students were paired up manually or by giving teachers the option to follow a dashboard's suggestions, which either paired students randomly or based on system-level knowledge estimates. Contrasting these policies was informed by prior research on tools for dynamically pairing students in APTA, including simulation studies and co-design sessions with teachers investigating human-AI collaboration for effective pairings and transitions between individual and collaborative learning [71, 35, 70]. During pairings, students had access to a chat interface through which they could exchange short text messages, which were logged. Teachers had access to a dashboard with estimated student skill estimates and the number of problems finished. When student pairs finished an assignment or were unpaired by the teacher, they returned to individual work. Thus, students only spent a portion of the time working collaboratively. Further information on the interplay of APTA, Lynnette, and Pair-Up, the interface used for supporting teachers in dynamically pairing students in this study, is described in Yang et al. [70].

3.2 Datasets

3.2.1 Tutor log data of student learning

We collected a pooled log data set of N = 185,641 transactions in the tutoring software (e.g., problem-solving step attempts, hint requests, and chat messages). All transactions were recorded via timestamped logs to DataShop [30]. The data set included N = 34,879 first attempts at problem-solving steps. A first attempt is the student's initial try at answering an equation-solving step in the tutor interface. As is common practice in modeling learning, we only sample first attempts because they represent students' initial response to solving linear equation-solving steps without Tutor support, with hints coded as incorrect attempts [32]. The data included 337 collaborative pairing episodes, which lasted an average of M = 25.38 mins (SD = 11.71). Each pairing had an average of M = 9.33 exchanged chat messages between Solver and Tutor (SD = 12.28). Our sample also included 3,058 chat messages by Solvers and Tutors following first attempts. There was an average of M = 6.02 messages in between first attempts (SD = 22.70). Messages before any first attempt in the system filtered out (2.7%).

[1] `https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId={5153,5549,5604}`

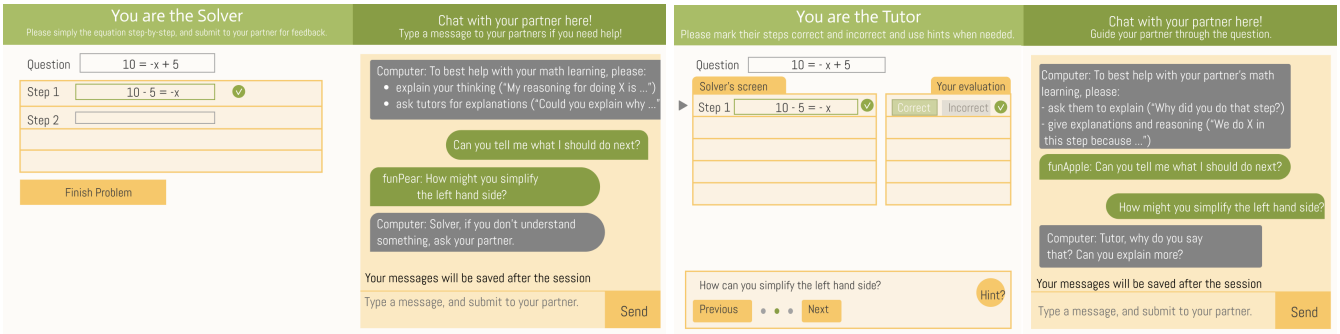

Figure 1: Interface of the collaborative tutoring system APTA from the Solver's (Upper) and Tutor's perspective (Lower).

3.2.2 Ground-truth labels of chat classifications

We developed a coding scheme for annotating dialog acts from student chat messages based on a content analysis of chat-based dialog during joint learning exercises. We follow methods by Wang et al. [66], who scrutinized chat data from peer-to-peer learning in a similar manner. We then adapted a coding scheme by Mawasi et al. [44], who identified peer assistance behaviors in middle school mathematics. Their framework had three primary labels for student discourse: minimal contribution, facilitative contribution, and constructive contribution. Based on prior work, dialog acts are expected to help learners most when they offer or elicit opportunities to learn [5], such as by prompting students to reflect, explain, and challenge their position [52]. In line with this reasoning, constructive contributions, which involve reasoning and explanation of content knowledge, are expected to be more helpful for learners than facilitative contributions, which offer less room for the tutored students to elaborate as they tend to focus more on asking for or giving out the answer. Minimal participation with little or no relevance to the content knowledge is likely least helpful for learners. The resulting coding scheme is exemplified in Table 1. Two coders separately coded N=1,000 randomly sampled chat messages based on the coding scheme. After three rounds of coding, the two coders achieved a satisfactory Cohen's κ of 0.70 [67] based on 101 observations and then proceeded to code all remaining chat messages.

3.3 Classification of chat messages

To learn a BERT model that can predict hand-coded chat message labels, we used transformer models that, unlike commercial large-language models like ChatGPT, are deployable and fine-tunable on regular personal computers. We used the open-source `bert-base-uncased` standard BERT model from the Python Transformers package. We fine-tuned BERT to predict the hand-coded label of a given chat message based on the text of the chat message and up to two chat messages preceding and following it (i.e., using up to five chat messages total). This decision was guided by prior work on classifying student-tutor dialog [37]. We pre-processed text to lowercase and removed punctuation.

From a sample of ground-truth 1,000 chat messages, we removed duplicates, 18 cases that could not be categorized, and 7 cases that had multiple labels across sentences, reducing the sample to $N = 890$ ground-truth chat message labels. We kept 20% of observations for a true holdout test set to evaluate model performance after cross-validation-based model selection. We used the remaining 80% of the data for model fine-tuning by evaluating the model after training epoch with batch size four. The number of epochs was determined such that model training was halted when no further improvements in accuracy on the training data were observed. The model was then evaluated on the holdout test set. For fine-tuning, we used a batch size of 16 with a learning rate of $5 * 10^{-5}$. The final model was applied to all remaining chat messages in our sample. Reproducible code for model training and data analyses is publicly available.[2]

3.4 Instructional factors analysis modeling

Instructional factors analysis (IFA) is a variation of additive factors modeling (AFM), a logistic regression model of students' gradual improvement over successive practice opportunities in the correctness of their attempts at solving problems step-by-step [32]. AFM assumes that the probability of getting steps right depends on the skills associated with the given problem-solving step. Therefore, AFM requires a series of problem-solving steps with an assessment of whether each is correct or incorrect and an underlying skill model annotating problem-solving steps with required cognitive operations. Then, AFM modeling estimates how much students improve in getting steps associated with certain skills right as a function of how many opportunities they had to apply a given skill to a new problem step instance (and receive feedback on their attempt, from which they can learn). These learning opportunities are observed steps in a practice problem (e.g., the next transformation of an equation) on which students receive as-needed feedback or instruction. Specifically, AFM models estimate *learning rates*: estimated improvements per opportunity in the probability of getting such steps correct. AFM further includes skill intercepts to estimate the initial difficulty of skills at the first opportunity and student intercepts to represent prior proficiency. We employed iAFM, a variation of AFM that additionally includes a student-level intercept for estimating prior proficiency using linear mixed modeling, which we also employ for IFA modeling [39]. In APTA, students can enter advanced steps into the system without performing all intermediate steps to solve the equation, which count as separate first attempts at skills related to all intermediate and performed steps in the log data.

[2] `https://github.com/conradborchers/peerchats-edm`

Table 1: coding scheme for chat messages based on sentence-level content analysis, including example messages for each category.

Category	Definition	Example messages
Minimal Contribution	Behavior that involves little to no domain content knowledge, e.g., greeting, confirming partner's identity, chatting or conversation related to usability and features of the tutoring systems	*"Who are you?"* *"OK"* *"I think it wants us to chat"*
Facilitative Contribution	Behaviors that involve domain content knowledge, and facilitate the collaboration by moving the conversation forward, but are of limited help on building transferable skills for the domain knowledge	*"This is wrong"* *"Type 2*x+2*-3+6=14"* *"You need a ="* *"Put that"*
Constructive Contribution	A statement involving reasoning and explanation of content knowledge. For example, answering a question with an explanation, correcting others with explanation, or asking a specific clarification question to help partner build transferable skills	*"Would you want me to explain?"* *"Tell me what you get when you divide by 4"* *"This is probably wrong?"* *"So the x has to be negative"*

The purpose of IFA modeling is to investigate whether different types of steps (for the same skill) yield different learning rates; more specifically, distinguish "instructional factors" that capture the presence or absence of instruction present in each step to model their influence on learning rates. To this end, IFA estimates separate learning rate parameters per type of instruction, counting opportunities to learn separately per instructional event type. IFA modeling describes learner data more accurately than AFM and Performance Factor Models (PFM) when multiple instructional interventions are present [14]. The larger the learning rate, the more effective the given instruction is considered to be [15]. An example data table is in Table 2.

Table 2: Example IFA data, with rows indicating learning opportunities for a unique skill and columns indicating cumulative opportunities related to a unique instructional factor. IFA estimates how much students improve after encountering a problem-solving step attempt with certain instruction.

Stud.	Skill	Opp. IF_1	Opp. IF_2	Opp. IF_3	Correct
Stu1	Sk1	1	0	0	0
Stu1	Sk1	1	1	0	0
Stu1	Sk1	1	2	0	1
Stu1	Sk2	0	0	1	0

The present study uses IFA to (a) compare learning rates of opportunities while learning with a peer tutor compared to working alone (RQ2) and (b) study how learning rates during collaborative learning (peer tutoring) might depend on the presence of the different kinds of chat messages (RQ3). For both purposes, we specify two model formulae of the iAFM model (with a general learning rate) and the IFA (with instructional factor-specific learning rates). The standard iAFM model is in Equation 1. The model infers the correctness of the first attempts at a problem-solving step ($Y_{correct}$) based on an individualized mixed-model student intercept representing students' initial proficiency ($\tau_{stud.}$), an intercept per skill representing initial skill difficulty (β_{skill}), and an opportunity count slope ($\beta_{opportunity}$).

$$Y_{correct} = \tau_{stud.} + \beta_{skill} + \beta_{opportunity} + \epsilon \quad (1)$$

The general IFA model formula is in Equation 2 and only different from Equation 1 in that it estimates separate learning rate slopes per opportunity related to a given number of instructional factors $\beta_{IFA_1} + \beta_{IFA_2} + ... + \beta_{IFA_n}$.

$$Y_{correct} = \tau_{stud.} + \beta_{skill} + \beta_{IFA_1} + \beta_{IFA_2} + ... + \beta_{IFA_n} + \epsilon \quad (2)$$

To answer RQ2, we compare an iAFM model featuring a standard opportunity count by skill (assuming no additional instructional factors) to an IFA model that distinguishes between paired and unpaired opportunities via a likelihood-ratio test. If that test is significant, students learn at significantly different rates based on whether they are paired or work alone. To answer RQ3, we similarly compare the same iAFM model as in RQ2 to a model that distinguishes opportunities by whether they were associated with no chat (including opportunities where students worked alone as opposed to collaboratively) or chats by the Tutor or the Solver with each being broken out by minimal vs. facilitative vs. constructive chats. Learning opportunities require a unique classification into minimal vs. facilitative vs. constructive. However, students could exchange multiple chat messages in between learning opportunities, for example, during repeated attempts. Therefore, we considered a learning opportunity to be related to the highest category of any chat in the dialog after each learning opportunity. For example, if a dialog included at least one constructive message by the Tutor and one facilitative message by the Solver, then the opportunity was assigned the instructional factors "Tutor Constructive" and "Solver Facilitative".

In all cases, we interpret the coefficients of the chosen model to compare learning across instructional factors. Learning rates correspond to the coefficients of the main effects of the opportunity counts in the model (e.g., number of opportunities related to a specific instructional factor; Table 2). We report learning rates expressed in odds ratios (OR), the factor by which two odds ($\frac{p}{1-p}$) of getting a first attempt in the tutoring software right is smaller or larger per opportunity. An OR of 1 results in a learning rate of 0, as ORs are centered around 1. For example, students improve per learning opportunity if the OR is significantly larger than 1, while an OR of 1 relates to constant performance. Notably, ORs can not be interpreted as frequencies; for example, correct attempts occur twice or half as often.

4. RESULTS

4.1 RQ1: Classification accuracy of student chat messages

Our first research question asks whether it is feasible to classify chat messages from the collaborative tutoring system into minimal, facilitative, and constructive transactions. Within our ground-truth data set of coded chat messages, 43.1% of messages were coded as minimal, while 43.7% were

facilitative and 13.1% constructive. Hence, students predominantly engaged in minimal conversational acts, such as establishing rapport or verifying their partner's identity, and in facilitative contribution, such as providing direct answers to their counterparts without additional guidance. During cross-validation, we did not observe further improvement in cross-validation accuracy at epoch four. Therefore, we stopped model training after four epochs and evaluated our classifier on the holdout test set (Table 3).

Table 3: Performance of final BERT classifier on the holdout test set.

Label	Precision	Recall	F_1	Support
Minimal	0.88	0.83	0.86	89
Facilitative	0.67	0.80	0.73	66
Constructive	0.80	0.52	0.63	23

The BERT classifier exhibited satisfactory performance across all three labels, with F_1 scores ranging from 0.86 (minimal) to 0.63 (constructive), which aligns with acceptable classification performance as suggested by similar studies (e.g., tutoring dialog act classification [55, 36, 56]). Constructive chat messages had a precision of 0.80 on the holdout test set, suggesting that the classifier is conservative in assigning that label. Overall, the final selected BERT classifier exhibited an accuracy of 0.78 (which is well above a majority-class classifier with 43.1% accuracy), a multi-class AUC of 0.91, a Cohen's κ of 0.63, and a macro average F_1 score of 0.74.

4.2 RQ2: Student learning rates when paired to a peer tutor vs. working alone

For RQ2, we compare an iAFM model featuring a standard opportunity count by skill to an IFA model that distinguishes between paired and unpaired opportunities via a likelihood-ratio test (see Equations 1 and 2). If that test is significant, students learn at significantly different rates based on whether they are paired or work alone. A likelihood-ratio test indicated that breaking opportunities into paired and unpaired opportunities did not significantly improve model fit ($\chi^2(1) = 1.52, p = .218$). Therefore, we do not find evidence for an overall difference in learning rates across paired and non-paired problem-solving. In both conditions, students significantly improved per opportunity with estimated learning rates of $OR = 1.04, CI_{95\%} = [1.02, 1.05], p < .001$ for paired and $OR = 1.03, CI_{95\%} = [1.02, 1.04], p < .001$ for unpaired opportunities. These results suggest that students' odds (i.e., $\frac{p_{correct}}{p_{incorrect}}$) of getting a first attempt in the tutoring software correct increased by around 3-4% per opportunity.

4.3 RQ3: Student learning differences by chat message type

Descriptively, Solvers sent considerably more chat messages than tutors. Out of 1,518 learning opportunities with chat messages, 1,416 included a message by the Solver (93.28%) and 123 by the Tutor (8.10%). Within the 1,416 chat-based opportunities with Solver messages, 302 (21.33%) conversations were classified as minimal, 566 (39.97%) facilitative, and 425 (30.01%) constructive. Within the 99 chat-based opportunities with Tutor messages, 18 (14.63%) were minimal, 39 (31.71%) facilitative, and 42 (34.25%) constructive. Overall, during students' collaborative learning episodes, only 11.41% of opportunities (i.e., first attempts at skills) were associated with chat messages by either Tutor or Solver, which breaks down to 0.92% for messages from the Tutor and 10.64% for messages from the Solver.

RQ3 compares an iAFM model featuring a standard opportunity count to an IFA model distinguishing between whether learning opportunities were associated with no chat (including opportunities during problem-solving alone) or chats by the Tutor or the Solver, with each being broken out by minimal vs. facilitative vs. constructive chats (see Equations 1 and 2). These models were still fit to all data, inclusive of students working alonse as opposed to pairs. The presence and types of chat messages as instructional factors significantly improved the IFA model fit ($\chi^2(6) = 56.37, p < .001$). Estimated model parameters of learning rates per instructional factor are in Table 4.

Table 4: Estimated learning rates by chat types with odds ratios (OR) of getting a problem-solving attempt right, excluding skill intercepts for brevity (N = 34,879).

Instructional Factor	OR	$CI_{95\%}$	p
No Message (N = 33,517)	1.02	[1.02, 1.03]	$< .001$
Tutor Minimal (N = 18)	1.92	[0.61, 6.09]	.268
Tutor Facilitative (N = 39)	1.97	[1.15, 3.38]	.014
Tutor Constructive (N = 42)	0.38	[0.24, 0.59]	$< .001$
Solver Minimal (N = 302)	1.40	[1.14, 1.72]	.001
Solver Facilitative (N = 566)	1.26	[1.10, 1.43]	.001
Solver Constructive (N = 425)	1.07	[0.93, 1.23]	.335

Solvers significantly improved per opportunity when conversations related to an opportunity included facilitative Tutor messages ($OR = 1.07, CI_{95\%} = [1.15, 3.38], p = .014$), minimal Solver messages ($OR = 1.40, CI_{95\%} = [1.14, 1.72], p = .001$) or facilitative Solver messages ($OR = 1.26, CI_{95\%} = [1.10, 1.43], p = .001$). Based on inspections of confidence intervals (CI), students also improved *more* per opportunities related to these conversations compared to opportunities with now messages, from which students still learned ($OR = 1.02, CI_{95\%} = [1.02, 1.03], p < .001$). Notably, students had significantly lower learning rate related to conversations inclusive of constructive Tutor messages ($OR = 0.38, CI_{95\%} = [0.24, 0.59], p < .001$), which warrants further inspection. All other learning rates of the considered instructional factors were not significantly different from a constant performance (i.e., such that there was no significant learning or decrease in performance).

4.4 Exploratory analysis

4.4.1 Solver Engagement

The results related to RQ3 indicate that students improved more per opportunity during problem-solving (while in the Solver role, helped by a peer Tutor) when sending minimal and facilitative. However, in the Tutor case, students only improved when conversations included facilitative chats. On the surface, it is not intuitive why Solvers should improve more per opportunity if their conversation is minimal. However, the reason might lie in what student-level differences relate to minimal Solver chats. Specifically, we hypothesized that a general engagement factor would relate to both

Solvers sending more minimal *and* facilitative chats, while that would not be the case for Tutors, which would have a specific tutoring style, tending to be *either* minimal or facilitative. We motivated that hypothesis based on a vast literature on the importance of affect and engagement for tutored and online learning [28, 65]. If that ad-hoc hypothesis is true, then Solvers should experience both minimal *and* facilitative opportunities but tend to experience *either* minimal *or* facilitative opportunities depending on the Tutor they were paired up with. We computed the total opportunity count for each chat message type to investigate the distribution of different kinds of Solver learning opportunities (Figure 2).

Figure 2 indicates that Solvers either experienced opportunities related to minimal or facilitative Tutor messages, neither, but rarely both. Solvers, on the other hand, when *generating* as opposed to receiving messages, tended to mix both types of messages more often. Yet, a large group of Solvers did not experience any chat-related opportunities, that is, none related to Solver messages (40.10%), tutor messages (89.09%), or even none at all (38.07%).

The positive relationship between Solvers' minimal and facilitative chat messages after first attempts and their learning rate (see Section 4.3) could possibly reflect a common cause (e.g., a high level of engagement with the collaborative tutoring system would lead to both more minimal chat messages and a higher learning rate) rather than a causal relation driving learning (where minimal chat message would be causing a higher learning rate, at odds to the definition of minimal chat messages stated above). We use the overall number of learning opportunities (i.e., first attempts at steps) per Solver to indicate overall engagement. When students have comparable amounts of time to engage with the tutoring system during classroom learning, opportunity counts are a better engagement measure of on-task learning than time-based measures of idle time in tutoring systems [53, 33, 31]. In line with the idea that Solver chats and learning rates are both related to engagement, the number of Solver messages and learning opportunities was significantly positively correlated (Spearman's $\rho = 0.13$, $p = .010$).

4.4.2 Properties of Constructive Tutor Dialog

Counterintuitively, constructive Tutor dialog was estimated to lower Solver learning rates. To further elucidate why, we qualitatively investigated conversations between Tutors and Solvers, including constructive messages from Tutors. We highlight informal themes in conversations between Solvers and Tutors via content analysis [45].

We identified three themes for why conversations classified as Tutor constructive related to opportunities did not help Solvers learn but potentially were counterproductive for their learning. First, we identified incorrect constructive messages by the Tutor, which could confuse students and reinforce misconceptions. An example was:

SOLVER: [*Minimal*] who are you

TUTOR: [*Minimal*] I'm <NAME>

SOLVER: [*Correct Input*] $\frac{4(x)}{4} = \frac{8}{4}$; $x = 2$

TUTOR: [*Constructive but Incorrect*] It would be 4 divide by 4 not x divide by 4

Second, constructive messages by the Tutor sometimes gave away the answer, which takes away the Solver's opportunity to learn from feedback, which is expected to relate to flat learning rates [22]. This issue was also exemplified by the fact that Solvers took fewer attempts at the problem-solving step to complete the step when the Tutor dialog included a constructive message ($M = 2.31$) compared to a facilitative message ($M = 3.09$). This was different for the constructive Solver dialog ($M = 3.10$) compared to the facilitative SOlver Dialog ($M = 2.51$). An example dialog after a learning opportunity with an incorrect attempt was:

TUTOR: [*Faciliative*] You need some help

TUTOR: [*Constructive*] It's x=5 not negative

Third, some constructive Tutor messages related to technical issues with the required ITS input syntax on the side of the Solver (e.g., missing a required equal sign). Such constructive dialog is unrelated to learning the skills required to solve the problem; hence, it is also not expected to help solvers improve. We share two examples:

TUTOR: [*Constructive*] You have to put the whole equation

TUTOR: [*Constructive*] You are missing a equal sign

4.5 Follow-up IFA model for error talk

Why was there little constructive dialog? Prior work on peer tutoring dialog motivates the hypothesis that productive talk may be more likely after errors [48]. Therefore, we explored another IFA model, which breaks out opportunity counts by whether a chat message occurred after a correct or incorrect first attempt in the system (Table 5).

Table 5: Estimated learning rates by chat types with odds ratios (OR) of getting a problem-solving attempt right, excluding skill intercepts for brevity ($N = 34{,}879$).

Instructional Factor	OR	$CI_{95\%}$	p
No Message	1.03	[1.02, 1.03]	$< .001$
Message Tutor Post-Correct	0.88	[0.63, 1.22]	.442
Message Tutor Post-Error	0.49	[0.09, 2.54]	.394
Message Solver Post-Correct	1.28	[1.17, 1.39]	$< .001$
Message Solver Post-Error	0.94	[0.79, 1.12]	.484

In this analysis, opportunities involving Tutor chat messages were not associated with significant Solver learning (p's $>$ 0.394; Table 5). However, opportunities related to messages sent by the Solver related to correct attempts ($OR = 1.28, CI_{95\%} = [1.17, 1.39], p < .001$) but not incorrect attempts ($OR = 0.94, CI_{95\%} = [0.74, 1.12], p = .484$) were associated with Solver improvement in the tutoring system. This finding does not align with the hypothesis that more helpful dialog is more likely after errors.

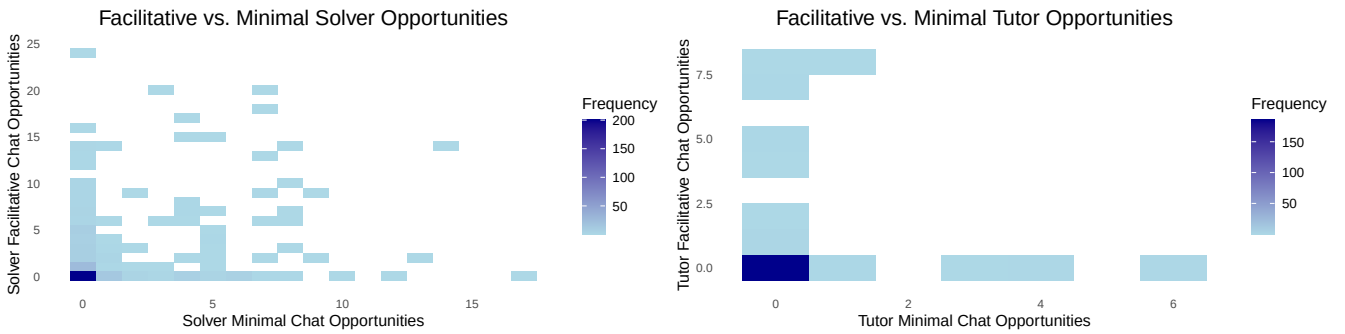

Figure 2: Distribution of total opportunity counts related to different chat message types (Solver message type on the left and Tutor message type on the right) by Solvers.

5. DISCUSSION

5.1 Summary of findings

Advances in NLP allow for the automated detection of dialog acts during peer tutoring. Yet, little work has integrated such detectors into learner modeling. The present study investigated the effectiveness of peer tutoring by comparing how much tutored students improve per practice opportunity in a collaborative tutoring system for middle school mathematics. The study further established the feasibility of classifying student chat messages using open-source transformer models (RQ1), whereby the classifier achieved an overall accuracy of nearly 80%, which is acceptable for automating dialog act classification as suggested by [36, 37].

RQ2 compared students' learning rates when paired compared to tutored problem solving alone. Students had no general advantage (or disadvantage) when working with a peer tutor. We did not detect a difference in learning rates across conditions. This finding does not align with prior estimations of learning rates in collaborative problem-solving, where learning rates were higher for procedural fraction problems [47]. One potential explanation for this finding is that only 11.41% of students' initial problem-solving attempts were followed by chat messages, and 10.64% of those opportunities included messages from Solvers only, not Tutors. More generally, the fact that the Solver did not show significantly greater learning from being peer-tutored is in line with prior findings in intelligent peer-tutoring systems that found that peer-tutors learn more than tutees when engaging in peer tutoring [63].

Yet, in the infrequent instances when messages were exchanged, students still generally learned differently depending on the *type* of message (RQ3). Our analysis suggests how working collaboratively may benefit learning, namely when Tutors and Solvers engage in facilitative dialog and when Solvers engage in minimal dialog, as we explore next.

5.2 Tutor facilitation and higher learning rates

While pairing did not generally increase student learning rate per opportunity (RQ2), we found that students exhibited larger learning rates when dialog included facilitative Tutor messages. Why did students learn more per opportunity when the Tutor employed facilitative but not minimal and constructive messages? Past work suggests that effective peer tutors offer or elicit opportunities to learn [5, 52]. Specifically, facilitative chat messages aim to move the collaboration forward, prompting the student to engage in problem-solving. This prompting could have led the students to reflect on their learning more often, effectively learning more per opportunity [6]. In contrast, minimal messages might not have given Solvers enough space to reflect on their learning, potentially taking the focus of the conversation off-topic. As an alternative explanation, Tutors employing facilitative chat messages could have given students a stronger sense of presence and caution to work on the problem rather than being off task, leading to higher learning rates as off-task behavior relates to close-to-flat learning rate [22]. Past work employing virtual agents to prevent students from gaming the system has shown similar results where off-task was greatly reduced through the mere presence of this virtual agent [7]. In line with this explanation, Solvers either experienced facilitation or minimal dialog through Tutors, such that the effectiveness of facilitative chat messages might not be a function of students' in-the-moment reflection on their learning through facilitative chat messages but rather through the presence of facilitative tutors. Field observations and think-aloud protocols bringing richer data on students' experience using the system in future work could distinguish between these explanations.

5.3 Solver learning related to engagement

Why were minimal Solver messages related to higher Solver learning rates? Our exploratory analysis suggests that engagement with the tutoring system, expected to relate to learning positively [22], could explain higher frequencies of Solver chat message opportunities and learning rates. Indeed, the number of chat messages and opportunities during problem-solving significantly positively correlated across Solvers. The hypothesis that engagement matters more than message types aligns with past MOOC research inferring learning gains from discussion posts [66]. The fact that only minimal and facilitative but not constructive chat messages significantly related to Solver's learning rates might be attributed to students using more constructive chat messages already coming in with more prior knowledge and having less to learn. However, future work is required to test this hypothesis. Alternatively, it could be that a higher level of chat engagement might reflect higher verbal skill, which prior work found to be associated with math learning [4]. A third alternative explanation is that the steps with constructive dialog are the ones where learning is harder, making

dialog engagement more likely. This selection effect would make errors more likely and learning rates lower, outweighing the beneficial effect of the dialog (if there is any). Similar selection effects have been investigated in the ITS literature on the negative correlation between help use (in the ITS) and pre-to-post learning gains [3]. Additional support for this interpretation comes from our follow-up analysis of whether students learned more per opportunity when messages were sent after correct vs. incorrect problem-solving step attempts. While prior work motivates the hypothesis productive talk is more common around errors, which would be expected to higher learning rates [48], our findings indicate that students learned more per opportunity after Solver messages after *correct*, not incorrect attempts. This finding, too, could be explained by a confound where students improve less as talk about errors relates to skills that students are more likely to make mistakes on.

5.4 Limitations in constructive tutor messages

Why was constructive chat dialog sent by the Tutor significantly related to lower learning rates by the Solver? Tutors were rarely engaged in constructive dialog (i.e., only after 2.77% of Solver learning opportunities), allowing us to qualitatively inspect the properties of Tutor dialog classified as constructive but related to lower learning rates. Indeed, what the employed BERT model classified as constructive messages might not always have been constructive for the Solver. Specifically, our inspection revealed that Tutors sometimes gave constructive but contextually incorrect advice to Solver, which might have caused confusion or even reinforced misconceptions that would explain lower learning rates (i.e., lower accuracy on subsequent attempts). This finding implies that future work should (a) incorporate information from ITS log data (i.e., whether a Tutor chat recommendation would lead to a correct attempt) into the prediction of the BERT model and (b) include training data examples of contextually incorrect constructive messages by Tutors into its sample. Indeed, prior work found similar integrations of log data into natural language for inference of self-regulated to be effective [20]. Similarly, nascent work found that the exclusion of problem-solving context in tutoring systems likely limits the accuracy of classifiers of self-regulation process stages based on natural language [72]. Considering ITS log data might help remediate other issues we observed with constructive Tutor dialog, log data could help detect when Tutors gave away the answer during dialog classified as constructive by the current classifier. Another possibility for improvement is to change our current definition of constructive dialog related to opportunities. In the present study, a dialog was classified as constructive if at least one message was constructive, which might be a threshold too liberal.

When subtracting the instances of factually incorrect help and ITS issues-related constructive dialog by Tutors, the already rare constructive chats become even rarer, limiting reliable estimations of constructive tutoring on learning with the present study's sample. If we assume that constructive chat messages are the gold standard for Tutoring, our results suggest that tutors in our sample did not provide much effective tutoring. One potential explanation for why students did not engage in constructive tutoring (or much tutoring at all) is that learning the content they are teaching while at the same time engaging in metacognitive learning of the task of tutoring is too high in cognitive load to happen concurrently, suggested by research on writing-to-learn [46]. When learning to tutor is already one learning objective of a tutoring system, engaging in dialog, specifically constructive dialog, is too cognitively demanding, and students may even resort to distracting chat messages, as described in Section 4.4.2. We discuss the implications of this interpretation next.

5.5 Implications

One key finding from this study is that Solvers generally improved when Tutors engaged in facilitative chats. At the same time, Tutors rarely engaged in chats, covering only 8.10% of Solvers' learning opportunities. This lack of chat engagement on the side of the Tutors could also likely explain why students did not improve more per learning opportunity overall when paired than when working alone (RQ2). Therefore, one potential revision to the current design of the collaborative tutoring system is to encourage Tutors to engage in more chat, especially facilitative chat. For example, the system could be redesigned to increase trust in the (anonymous) chat partner, as prior work showed associations between trust and engagement in learning management systems [25]. An alternative solution could be to redesign the ITS to help Tutors grade the steps of the Solver more automatically once Tutors have demonstrated mastery of the relevant steps to free up more resources to tutor through chat. Another element that may help Tutors become more effective is to align the Tutor's incentive with the learning objectives of the Solver, which prior research delineates as one success factor in effective peer tutoring [27]. An example of fostering incentive alignment would be having the Tutor's grades depend on the learner success of the Solver. However, such policies might be rather stringent in the middle school classroom, if not norm-breaking. A more toned-down version of incentive alignment that could be explored in future work would be to reward Tutors more by tutoring effectively, such as using skill bars as formative assessments where Tutors watch their progress in learning how to tutor Solvers effectively. A third potential design change is to deploy our classification model for chat messages in real-time to detect when Tutors send excessive amounts of minimal chat messages and prompt Tutors to use facilitative or constructive messages, potentially giving them example messages to send.

Further, Tutors gave little tutoring through messaging overall. This could be because the system is concurrently teaching the Tutors how to tutor, while the Solvers primarily receive feedback on their problem-solving attempts similar to working alone [63]. The rarity of constructive dialog on the side of the Tutor might relate to the excessive cognitive load of Tutors learning how to tutor and mastering the tutored math content simultaneously. Therefore, one potential design implication of the present study is that Tutors may require more upfront training on how to tutor with clearly designed learning objectives and assessments while tutoring content knowledge that they have already mastered, as opposed to content they are still learning. Specifically, students in classrooms who are quick to master relevant skills during individualized problem-solving could be adaptively paired with students who have not mastered their skills. In this study, most students were paired based on pairing of advanced students with less advanced students based on the

teacher's judgment of dashboard recommendations [35, 70], which leaves room to investigate the benefits of mastery-based pairings in future work. Upfront training of the Tutors has been shown to improve tutoring practice, which is expected to improve the learning of tutees [59]. Training could include a tutor effectiveness score via live analytics and simplified grading of Solver steps to free up cognitive resources (and feedback based on classifications of chat messages).

5.6 Limitations and future work

We see three limitations to this study that may guide future work. First, our current classification algorithm, although following best practices from prior work on dialog act classification [36], was trained on an unbalanced data set. This might cause the classifier to prioritize learning the patterns from the majority class, leading to a relatively low recall rate of constructive messages [38]. Downstream applications might benefit more from balancing precision and recall for constructive messages. For example, a critical consideration in training peer tutors is detecting constructive messages to give feedback. In future work, applications might benefit from more liberal thresholds to avoid students being frustrated when generating a constructive message. Future work could also experiment with in-depth cross-validations of hyperparameters for BERT model fine-tuning, which we did not perform due to time constraints during model training.

A second limitation is that our present sample is limited to mathematics. As past work, compared to the present study, observed at least *some* benefits in learning rates from collaborative problem-solving in tutoring systems for procedural fraction problems [47], more work is needed to replicate our present methodology of estimating learning rates by chat message type to other subject domains. Future work could also investigate if our trained model for classifying peer tutor messages is sufficiently accurate to other subject domains or if adjustments to our current coding scheme need to be made, which was adapted from a coding scheme of peer assistance behaviors in middle school mathematics [44].

A third limitation is that given low chat activity, specifically for Tutors, our modeling might not detect potential relationships between messages and learning rates due to low statistical power. Future work could consider extensions of IFA modeling to increase power by relaxing the assumption that each learning opportunity needs to be associated with one instructional factor (e.g., type of learning opportunity) only. Next to assigning each dialog multiple instructional factors if multiple types of chats are present in the dialog related to a learning opportunity, future work could consider modeling performance on each attempt, including re-attempts at the problem-solving step, to understand student learning [9].

6. SUMMARY AND CONCLUSIONS

The present study contributes novel evidence of how different chat messages in intelligent collaborative learning systems relate to tutored students' learning rates. We establish a reliable classifier with an accuracy of close to 80% to classify student chat messages during peer tutoring into minimal, facilitative, and constructive messages. Low chat engagement of Tutors accounted for only 8% of the learning opportunities associated with chat messages during pairings. However, when Tutors messaged, facilitative dialog was associated with higher Solver learning rates. Further, we found a relationship between Solver engagement and their learning rate. Finally, Tutor dialog classified as constructive was even rarer (3%) and related to lower Solver learning rates, potentially due to reinforcing incorrect solutions, giving away answers, or being unrelated to the taught content. An excessive cognitive load of learning to tutor while simultaneously learning math content is the most viable reason Tutors could not engage with Solvers more productively using chats. In closing, our findings also tell a cautionary tale, recommending future efforts to integrate problem-solving context into chat prediction, such as detecting contextually incorrect peer tutor messages. We encourage research to replicate our methodology of combining message classification with learning rate modeling to understand collaborative learning.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1822861.

7. REFERENCES

[1] M. Abdelshiheed, J. K. Jacobs, and S. K. D'Mello. Aligning tutor discourse supporting rigorous thinking with tutee content mastery for predicting math achievement. In *Proceedings of the 25th International Conference on Artificial Intelligence in Education (AIED'24)*, Recife, Brazil, 2024.

[2] A. Abulimiti, C. Clavel, and J. Cassell. When to generate hedges in peer-tutoring interactions. In *24th Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2023.

[3] V. Aleven, I. Roll, B. M. McLaren, and K. R. Koedinger. Help helps, but only so much: Research on help seeking with intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 26:205–223, 2016.

[4] H. Almoubayyed, S. E. Fancsali, and S. Ritter. Generalizing predictive models of reading ability in adaptive mathematics software. In *Proceedings of the 16th International Conference on Educational Data Mining*, 2023.

[5] C. Asterhan, S. Clarke, and L. Resnick. Socializing intelligence through academic talk and dialogue. *Socializing Intelligence Through Academic Talk and Dialogue*, pages 1–480, 2015.

[6] R. K. Atkinson and A. Renkl. Interactive example-based learning environments: Using interactive elements to encourage effective processing of worked examples. *Educational Psychology Review*, 19:375–386, 2007.

[7] R. S. d. Baker, A. T. Corbett, K. R. Koedinger, S. Evenson, I. Roll, A. Z. Wagner, M. Naim, J. Raspat, D. J. Baker, and J. E. Beck. Adapting to when students game an intelligent tutoring system. In *Intelligent Tutoring Systems: 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, 2006. Proceedings 8*, pages 392–401. Springer, 2006.

[8] C. Borchers, P. F. Carvalho, M. Xia, P. Liu, K. R. Koedinger, and V. Aleven. What Makes Problem-Solving Practice Effective? Comparing Paper and AI Tutoring. In *European Conference on Technology Enhanced Learning*, pages 44–59. Springer,

2023.

[9] C. Borchers, J. Zhang, R. S. Baker, and V. Aleven. Using think-aloud data to understand relations between self-regulation cycle characteristics and student performance in intelligent tutoring systems. In *LAK24: 14th International Learning Analytics and Knowledge Conference*, 2024.

[10] K. Boyer, E. Y. Ha, R. Phillips, M. Wallis, M. Vouk, and J. Lester. Dialogue act modeling in a complex task-oriented domain. In *Proceedings of the SIGDIAL 2010 Conference*, pages 297–305, 2010.

[11] K. Boyer, R. Phillips, M. Wallis, M. Vouk, and J. Lester. Learner characteristics and feedback in tutorial dialogue. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–61, 2008.

[12] M. Burton, P. Brna, and R. Pilkington. Clarissa: a laboratory for the modelling of collaboration. *International Journal of Artificial Intelligence in Education*, 11(2):79–105, 2000.

[13] P. Chejara, L. P. Prieto, M. J. Rodriguez-Triana, R. Kasepalu, A. Ruiz-Calleja, and S. K. Shankar. How to build more generalizable models for collaboration quality? lessons learned from exploring multi-context audio-log datasets using multimodal learning analytics. In *LAK23: 13th International Learning Analytics and Knowledge Conference*, pages 111–121, 2023.

[14] M. Chi, K. Koedinger, G. Gordon, and P. Jordan. Instructional factors analysis: A cognitive model for multiple instructional interventions. In *Proceedings of the 4th International Conference on Educational Data Mining (EDM)*, 2011.

[15] N. Diana, J. Stamper, and K. Koedinger. An instructional factors analysis of an online logical fallacy tutoring system. In *Artificial Intelligence in Education: 19th International Conference, AIED 2018, London, UK, June 27–30, 2018, Proceedings, Part I 19*, pages 86–97. Springer, 2018.

[16] S. D'Mello, A. Olney, C. Williams, and P. Hays. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of Human-Computer Studies*, 70(5):377–398, 2012.

[17] A. Dood, B. Winograd, S. Finkenstaedt-Quinn, A. Gere, and G. Shultz. Peerbert: Automated characterization of peer review comments across courses. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 492–499, 2022.

[18] T. V. Earle-Randell, J. B. Wiggins, J. M. Ruiz, M. Celepkolu, K. E. Boyer, C. F. Lynch, M. Israel, and E. Wiebe. Confusion, conflict, consensus: Modeling dialogue processes during collaborative learning with hidden markov models. In *International Conference on Artificial Intelligence in Education*, pages 615–626. Springer, 2023.

[19] V. Echeverria, K. Holstein, J. Huang, J. Sewall, N. Rummel, and V. Aleven. Exploring human–ai control over dynamic transitions between individual and collaborative learning. In *Addressing Global Challenges and Quality Education: 15th European Conference on Technology Enhanced Learning, EC-TEL 2020, Heidelberg, Germany, September 14–18, 2020, Proceedings 15*, pages 230–243. Springer, 2020.

[20] Y. Fan, M. Rakovic, J. van Der Graaf, L. Lim, S. Singh, J. Moore, I. Molenaar, M. Bannert, and D. Gašević. Towards a fuller picture: Triangulation and integration of the measurement of self-regulated learning based on trace and think aloud data. *Journal of Computer Assisted Learning*, 39(4):1303–1324, 2023.

[21] M. Flor, S.-Y. Yoon, J. Hao, L. Liu, and A. von Davier. Automated classification of collaborative problem solving interactions in simulated science tasks. In *Proceedings of the 11th workshop on innovative use of NLP for building educational applications*, pages 31–41, 2016.

[22] Y. Gong, J. E. Beck, N. T. Heffernan, and E. Forbes-Summers. The fine-grained impact of gaming (?) on learning. In *Intelligent Tutoring Systems: 10th International Conference, ITS 2010, Pittsburgh, PA, USA, June 14-18, 2010, Proceedings, Part I 10*, pages 194–203. Springer, 2010.

[23] S. Hobert and R. M. V. Wolff. Say hello to your new automated tutor – a structured literature review on pedagogical conversational agents. In *Proceedings of International Conference on Wirtschaftsinformatik*, pages 301–314, Siegen, Germany, 2019.

[24] Y. Huang, N. G. Lobczowski, J. E. Richey, E. A. McLaughlin, M. W. Asher, J. M. Harackiewicz, V. Aleven, and K. R. Koedinger. A general multi-method approach to data-driven redesign of tutoring systems. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pages 161–172, 2021.

[25] S. N. Ismail, S. Hamid, M. Ahmad, A. Alaboudi, and N. Jhanjhi. Exploring students engagement towards the learning management system (lms) using learning analytics. *Computer Systems Science & Engineering*, 37(1), 2021.

[26] D. Johnson and R. Johnson. Cooperative learning and achievement. cooperative learning: theory and research. *New York: Praeger*, pages 23–37, 1990.

[27] D. W. Johnson, R. T. Johnson, and M. B. Stanne. Cooperative learning methods: A meta-analysis. 2000.

[28] A. Joshi, D. Allessio, J. Magee, J. Whitehill, I. Arroyo, B. Woolf, S. Sclaroff, and M. Betke. Affect-driven learning outcomes prediction in intelligent tutoring systems. In *2019 14th ieee international conference on automatic face & gesture recognition (fg 2019)*, pages 1–5. IEEE, 2019.

[29] S. Katz, P. Albacete, I.-A. Chounta, P. Jordan, B. M. McLaren, and D. Zapata-Rivera. Linking dialogue with student modelling to create an adaptive tutoring system for conceptual physics. *IJAIED*, 31(3):397–445, 2021.

[30] K. R. Koedinger, R. S. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the edm community: The pslc datashop. *Handbook of Educational Data Mining*, 43:43–56, 2010.

[31] K. R. Koedinger, P. F. Carvalho, R. Liu, and E. A. McLaughlin. An astonishing regularity in student learning rate. *Proceedings of the National Academy of Sciences*, 120(13):e2221311120, 2023.

[32] K. R. Koedinger, S. D'Mello, E. A. McLaughlin, Z. A. Pardos, and C. P. Rosé. Data mining and education. *Wiley Interdisciplinary Reviews: Cognitive Science*, 6(4):333–353, 2015.

[33] K. R. Koedinger, J. Kim, J. Z. Jia, E. A. McLaughlin, and N. L. Bier. Learning is not a spectator sport: Doing is better than watching for learning from a mooc. In *Proceedings of the Second ACM Conference on Learning@Scale*, pages 111–120, 2015.

[34] A. M. Latham, K. A. Crockett, D. A. McLean, B. Edmonds, and K. O'shea. Oscar: An intelligent conversational agent tutor to estimate learning styles. In *International conference on fuzzy systems*, pages 1–8. IEEE, 2010.

[35] L. Lawrence, B. Guo, V. Echeverria, Z. Kang, V. Bathala, C. Li, W. Huang, V. Aleven, and N. Rummel. Co-designing ai-based orchestration tools to support dynamic transitions: Design narratives through conjecture mapping. *Proceedings of the International Conference on Computer-Supported Collaborative Learning (CSCL)*, 15, 2022.

[36] J. Lin, S. Singh, L. Sha, W. Tan, D. Lang, D. Gašević, and G. Chen. Is it a good move? mining effective tutoring strategies from human–human tutorial dialogues. *Future Generation Computer Systems*, 127:194–207, 2022.

[37] J. Lin, W. Tan, L. Du, W. Buntine, D. Lang, D. Gašević, and G. Chen. Enhancing educational dialogue act classification with discourse context and sample informativeness. *IEEE TLT*, 2023.

[38] J. Lin, W. Tan, N. D. Nguyen, D. Lang, L. Du, W. Buntine, R. Beare, G. Chen, and D. Gašević. Robust educational dialogue act classifiers with low-resource and imbalanced datasets. In *AIED23*, pages 114–125. Springer, 2023.

[39] R. Liu and K. R. Koedinger. Towards reliable and valid measurement of individualized student parameters. In *Proceedings of the 10th International Conference on Educational Data Mining (EDM)*, 2017.

[40] Y. Long, K. Holstein, and V. Aleven. What exactly do students learn when they practice equation solving? refining knowledge components with the additive factors model. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, pages 399–408, 2018.

[41] Y. Ma, M. Celepkolu, and K. E. Boyer. Detecting impasse during collaborative problem solving with multimodal learning analytics. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 45–55, 2022.

[42] J. Malmberg, M. Saqr, H. Järvenoja, and S. Järvelä. How the monitoring events of individual students are associated with phases of regulation: A network analysis approach. *Journal of learning analytics*, 9(1):77–92, 2022.

[43] N. Matsuda, E. Yarzebinski, V. Keiser, R. Raizada, G. J. Stylianides, W. W. Cohen, and K. R. Koedinger. Learning by teaching simstudent–an initial classroom baseline study comparing with cognitive tutor. In *Artificial Intelligence in Education: 15th International Conference, AIED 2011, Auckland, New Zealand, June 28–July 2011 15*, pages 213–221. Springer, 2011.

[44] A. Mawasi, I. Ahmed, E. Walker, S. Wang, Z. Marasli, A. Whitehurst, and R. Wylie. Using design-based research to improve peer help-giving in a middle school math classroom. 2020.

[45] K. A. Neuendorf. *The content analysis guidebook*. Sage, 2017.

[46] M. Nückles, J. Roelle, I. Glogger-Frey, J. Waldeyer, and A. Renkl. The self-regulation-view in writing-to-learn: Using journal writing to optimize cognitive load in self-regulated learning. *Educational Psychology Review*, 32:1089–1126, 2020.

[47] J. Olsen, V. Aleven, and N. Rummel. Statistically modeling individual students' learning over successive collaborative practice opportunities. *Journal of Educational Measurement*, 54(1):123–138, 2017.

[48] J. Olsen, N. Rummel, and V. Aleven. Finding productive talk around errors in intelligent tutoring systems. In *Proceedings of the International Conference on Computer Supported Collaborative Learning (CSCL)*, 2015.

[49] J. K. Olsen, K. Sharma, N. Rummel, and V. Aleven. Temporal analysis of multimodal data to predict collaborative learning outcomes. *British Journal of Educational Technology*, 51(5):1527–1547, 2020.

[50] J. Paladines and J. Ramirez. A systematic literature review of intelligent tutoring systems with dialogue in natural language. *IEEE Access*, 8:164246–164267, 2020.

[51] S. L. Pugh, S. K. Subburaj, A. R. Rao, A. E. Stewart, J. Andrews-Todd, and S. K. D'Mello. Say what? automatic modeling of collaborative problem solving skills from student speech in the wild. In *Proceedings of the 14th International Conference on Educational Data Mining (EDM)*, 2021.

[52] L. B. Resnick, C. S. Asterhan, and S. N. Clarke. Accountable talk: Instructional dialogue that builds the mind. *Geneva, Switzerland: The International Academy of Education (IAE) and the International Bureau of Education (IBE) of the UNESCO*, 2018.

[53] S. Ritter, A. Joshi, S. Fancsali, and T. Nixon. Predicting standardized test scores from cognitive tutor interactions. In *Proceedings of the Sixth International Conference on Educational Data Mining (EDM)*, 2013.

[54] F. J. Rodríguez, K. M. Price, and K. E. Boyer. Exploring the pair programming process: Characteristics of effective collaboration. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pages 507–512, 2017.

[55] V. Rus, N. Maharjan, L. J. Tamang, M. Yudelson, S. Berman, S. E. Fancsali, and S. Ritter. An analysis of human tutors' actions in tutorial dialogues. In *The Thirtieth International Flairs Conference*, 2017.

[56] K. Stasaski, K. Kao, and M. A. Hearst. Cima: A large open access dialogue dataset for tutoring. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64, 2020.

[57] S. Strauß and N. Rummel. Promoting regulation of equal participation in online collaboration by combining a group awareness tool and adaptive

prompts. but does it even matter? *International Journal of Computer-Supported Collaborative Learning*, 16:67–104, 2021.

[58] K. Thaker, P. Carvalho, and K. Koedinger. Comprehension factor analysis: Modeling student's reading behaviour: Accounting for reading practice in predicting students' learning in moocs. In *LAK19: 9th International Learning Analytics and Knowledge Conference*, pages 111–115, 2019.

[59] D. Thomas, X. Yang, S. Gupta, A. Adeniran, E. Mclaughlin, and K. Koedinger. When the tutor becomes the student: Design and evaluation of efficient scenario-based lessons for tutors. In *LAK23: 13th International Learning Analytics and Knowledge Conference*, pages 250–261, 2023.

[60] S. Ubani, R. Nielsen, and H. Li. Detecting exclusive language during pair programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15964–15971, 2023.

[61] A. K. Vail and K. E. Boyer. Identifying effective moves in tutoring: On the refinement of dialogue act annotation schemes. In *International Conference on Intelligent Tutoring Systems*, pages 199–209. Springer, 2014.

[62] E. Walker, N. Rummel, and K. R. Koedinger. Designing automated adaptive support to improve student helping behaviors in a peer tutoring activity. *International Journal of Computer-Supported Collaborative Learning*, 6:279–306, 2011.

[63] E. Walker, N. Rummel, and K. R. Koedinger. Adaptive intelligent support to improve peer tutoring in algebra. *International Journal of Artificial Intelligence in Education*, 24:33–61, 2014.

[64] Q. Wang. Design and evaluation of a collaborative learning environment. *Computers & Education*, 53(4):1138–1146, 2009.

[65] X. Wang, M. Wen, and C. P. Rosé. Towards triggering higher-order thinking behaviors in moocs. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 398–407, 2016.

[66] X. Wang, D. Yang, M. Wen, K. Koedinger, and C. P. Rosé. Investigating how student's cognitive behavior in mooc discussion forums affect learning gains. In *Proceedings of the 8th International Conference on Educational Data Mining (EDM)*, 2015.

[67] M. J. Warrens. Five ways to look at cohen's kappa. *Journal of Psychology & Psychotherapy*, 5, 2015.

[68] A. Weerasinghe, A. Mitrovic, M. Van Zijl, and B. Martin. Evaluating the effectiveness of adaptive tutorial dialogues in database design. In *Proceedings of the 18th International Conference on Computers in Education*, pages 33–40, 2010.

[69] A. F. Wise, S. Knight, and S. B. Shum. Collaborative learning analytics. *International Handbook of Computer-Supported Collaborative Learning*, pages 425–443, 2021.

[70] K. B. Yang, V. Echeverria, Z. Lu, H. Mao, K. Holstein, N. Rummel, and V. Aleven. Pair-up: Prototyping human-ai co-orchestration of dynamic transitions between individual and collaborative learning in the classroom. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2023.

[71] K. B. Yang, V. Echeverria, X. Wang, L. Lawrence, K. Holstein, N. Rummel, and V. Aleven. Exploring policies for dynamically teaming up students through log data simulation. In *Proceedings of the 14th International Conference on Educational Data Mining (EDM)*, 2021.

[72] J. Zhang, C. Borchers, V. Aleven, and R. S. Baker. Using large language models to detect self-regulated learning in think-aloud protocols. In *Proceedings of the 17th International Conference on Educational Data Mining (EDM)*, 2024.