Prompting as Panacea? A Case Study of In-Context Learning Performance for Qualitative Coding of Classroom Dialog

Ananya Ganesh, Chelsea Chandler, Sidney D'Mello, Martha Palmer, Katharina von der Wense
University of Colorado Boulder
Boulder, CO, 80309
Ananya.Ganesh@colorado.edu

ABSTRACT

One of the areas where Large Language Models (LLMs) show promise is for automated qualitative coding, typically framed as a text classification task in natural language processing (NLP). Their demonstrated ability to leverage *in-context learning* to operate well even in data-scarce settings poses the question of whether collecting and annotating large-scale data for training qualitative coding models is still beneficial. In this paper, we empirically investigate the performance of LLMs designed for use in prompting-based in-context learning settings, and draw a comparison to models trained with task-specific annotated data, specifically for tasks involving qualitative coding of classroom dialog. Compared to other domains where NLP bench-marking studies are typically situated, classroom dialog is much more natural and therefore variable and complex. Moreover, tasks in this domain are nuanced, theoretically grounded and require a deep understanding of the conversational context. We provide a comprehensive evaluation across five datasets, including tasks such as talk move prediction and collaborative problem solving skill identification. Our findings show that task-specific fine-tuning strongly outperforms in-context learning, underscoring the ongoing need for high-quality annotated training datasets.

Keywords

large language models, natural language processing, qualitative coding, classroom dialog

1. INTRODUCTION

In recent years, the proliferation of Natural Language Processing (NLP), Artificial Intelligence (AI), and Large Language Models (LLMs) has revolutionized various facets of educational technology, from the development of conversational tutors to automated grading systems, significantly impacting student learning experiences and instructional methodologies [4, 33, 20]. LLMs, particularly those that can be used off-the-shelf with minimal to no training such as the GPT [22] and LLaMa [31] models are slowly being adopted as the de facto models for text generation tasks such as generating hints [25], providing feedback to students [18], or assisting teachers [33]. More recently, LLMs have gained attention as an alternative to "traditional" NLP models for automated qualitative coding tasks [34]. However, their feasibility for coding tasks, especially for challenging constructs commonly found in the educational domain, has yet to be systematically investigated.

Automated qualitative coding problems are typically formalized as classification tasks in NLP. That is, given text (such as student conversation or writing), the task is to predict the most likely label from a pre-defined set of classes (such as if an utterance is a question). Classification models have until recently been developed through the pre-training–fine-tuning paradigm: pre-trained language models such as BERT [7], trained on large web-based corpora are *fine-tuned*, or undergo task-specific training on human-annotated data. While these models leverage the rich representations of language learned in the pre-training stage to understand meaning, they require several examples of each class to learn to distinguish between them, ultimately necessitating datasets with thousands of examples. Acquiring such large datasets incurs substantial costs, as trained experts must invest time and effort into the annotation process. On the other hand, the ability of LLMs to learn to solve complex tasks using very few or no examples ostensibly provides a way to bypass the expensive data collection process. LLMs achieve this through the mechanism of in-context learning, a process that involves interacting with LLMs through natural language instructions, without any training.

This ability has been demonstrated on NLP benchmarks that challenge the language understanding abilities of models [28]. Some of these benchmarks even test for domain knowledge and reasoning on topics such as physics or medicine [21, 12]. Despite such rigorous testing, applying these models to qualitative coding, particularly in education, should still be treated with caution for the following reasons: 1) benchmark tasks tend to be well-defined and sometimes shallow, in contrast to the nuanced, theoretically-motivated frameworks that drive qualitative coding, 2) benchmark tasks may be highly similar to tasks that some LLMs are explicitly trained on (e.g., question answering); and 3) benchmark datasets, particularly if openly available online, may have

A. Ganesh, C. Chandler, S. D'Mello, M. Palmer, and K. Kann. Prompting as panacea? a case study of in-context learning performance for qualitative coding of classroom dialog. In B. Paaßen and C. D. Epp, editors, *Proceedings of the 17th International Conference on Educational Data Mining*, pages 835–843, Atlanta, Georgia, USA, July 2024. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.12729966

already been seen by LLMs in their extensive pre-training stage. Therefore, in this paper, we explore how prompting off-the-shelf generalist LLMs performs on nuanced, real world tasks that are unlikely to have been seen in training.

We choose the problem of analyzing student dialog as a case study for our investigation. This task is significant, and critically impacts multiple stakeholders: it can inform pedagogical practices by enhancing educators' and learning scientists' understanding of student experience [10], and can steer learning tools like tutoring systems through specific properties of student utterances like talk moves [30]. Additionally, utterances spoken naturally by students tend to be noisy, incomplete and contextual, making them challenging for NLP models typically trained on sterilized text. Physical classrooms also include complications of multi-party interactions and automatic speech recognition (ASR) errors. To ensure a range of these phenomena, we provide a broad perspective of LLM performance on nine qualitative coding tasks across five datasets based in education, ranging in size from 2500 to 50k examples. Focusing on the question of whether annotated data still plays an important role in automated qualitative coding, we compare the performance of in-context learning in LLMs to task-specific classification models. Finally, based on our investigations, we provide recommendations for choosing models for student dialog analysis when cost, data, and performance need to be balanced.

2. RELATED WORK

LLMs in Educational Applications: Since their rise to fame, LLMs have readily been adopted into applications to support learning [23, 19, 5]. The capacity of LLMs to *generate* unique, coherent and meaningful responses conditioned on prompts, even with limited or no task-specific adaptation is the capability that is most utilized in these applications. GPT-4 is famously used in Khanmigo [26] to provide guidance in the form of hints and questions as students work through lessons. While promising, some studies show that effectiveness of such feedback could be improved [25]. Other works point out how LLM-generated feedback directed at teachers could help with teacher coaching [33], and how reframings of teacher feedback could lead to improvement in student's mindsets [11].

LLMs for Qualitative Coding: More relevant to our work is the performance of LLMs on qualitative coding tasks. [34] explore ChatGPT's performance at categorizing student feedback found in the comments of MOOC open-access videos. On a set of 200 human-annotated examples, they found that model-human agreement falls short of human-human agreement overall, but on less complex categories like expressing gratitude, the model performs at par with humans. Others have demonstrated the success of LLMs for qualitative coding broadly in domains like computational social science [37] and medicine [21]. Researchers also acknowledge that in any task, particularly in the zero-shot setting, LLMs are sensitive to the input prompt [27], which is especially troublesome for non AI experts who may struggle to systematically explore effective prompt designs [36].

Due to these reasons and the high variability in LLM performance based on task complexity, [24] argue that LLMs must be validated using a human annotation process, where both humans and LLMs are provided access to the same guidelines or codebook. LLMs have also been explored as a collaborator, rather than a replacement, to human coders, such as in the LLM-in-the-loop framework [4].

Comparisons to Task-Specific Models: Similar to our work, prior research has investigated LLM performance in comparison to models trained on task-specific data. [15] assessed the classification performance of RoBERTa [17] and GPT-3 [2], both fine-tuned on training data, in categorizing teacher talk moves. The fine-tuned GPT-3 model had a higher F1 score but poorer recall than RoBERTa on a majority of the categories. However, on the class with the smallest number of training examples, GPT-3 performed significantly worse, contrary to widely-held beliefs about the superior few-shot learning abilities of recent LLMs. Additionally, [21] found that a carefully engineered few-shot learning approach with GPT-4 outperformed a specialized task-specific model named Med-Palm-2 on medical question answering.

However, depending on the task and complexity, task-specific fine-tuning has outperformed prompting methods. [9] compared fine-tuned RoBERTa models to prompting ChatGPT on the task of predicting dialog behaviors, such as showing empathy or contradicting oneself. ChatGPT performed slightly worse than fine-tuned models on three out of five behavioral categories. [37] also found that fine-tuned RoBERTa models outperformed zero-shot LLM prompting in almost all cases in a computational social science task.

Our novel contributions are: (i) we conduct a broad analysis of LLM performance on educational qualitative coding that is less restricted to specific datasets or phenomena (ii) we evaluate across multiple LLMs, prompting strategies, and fine-tuning strategies, covering closed and open-source models (iii) we compare strategies with various amounts of training data, investigating whether collecting annotated data is still a valuable endeavor in the age of LLMs.

3. DATASETS AND TASKS

The datasets that we use are either open access, or have been extensively described in prior publications. We provide a brief overview here, and include information about label distributions in the appendix.

Collaborative Problem Solving: We used two CPS datasets – one collected during a block programming challenge (Minecraft) and another from a middle school science curriculum (Sensor Immersion). The three CPS facets coded in these datasets are *constructing shared knowledge*, *negotiation / coordination*, and *maintaining team function*. Utterances are coded with binary indicators for each label individually, with 1 indicating that the CPS facet is present in the utterance. For more information, we refer the reader to [29] and [3].

Student Talk Move Prediction: We use the Talk Move dataset of K-12 mathematics lesson transcripts, containing teacher and student speech [30]. For this work, we focused on classifying *student* talk moves, however models preceding teacher utterances were given as context. Student talk moves include *relating to another student*, *asking for more information*, *making a claim*, *providing evidence*, or *None*.

NCTE: Identifying Student Reasoning The National Center for Teacher Effectiveness (NCTE) dataset consists of anonymized transcripts of classroom instruction from US elementary-school mathematics classrooms. We focus on the binary task of identifying if student utterances contain reasoning. More details can be found in [6].

CIMA: Identifying Student Actions Conversational Instruction with Multi-responses and Actions (CIMA) is an open-access corpus of crowdsourced one-on-one tutoring dialog. In contrast to our other datasets, utterances are typed and not spoken or transcribed. We investigate the task of student action classification, predicting whether student responses are a *Guess, Question, Affirmation* or *Other.* The labels are self-reported by the crowdworker producing the utterance.

We divide our datasets into subsets for training, development and testing according to a 70%/15%/15% ratio. A notable problem here, as with most educational dialog datasets, is label imbalance, which has been recognized in prior work [16]. However, strategies for mitigating label imbalance are out of scope for the present work.

4. METHODS

We compare LLMs used for classification in a *prompting* setting with models *trained* on task-specific annotated data. Below, we provide an overview of the models and methods used for prompting and fine-tuning.

4.1 Prompting

GPT-4 and Mistral: GPT-4 [22] is a Transformer-based language model developed by OpenAI. The model is pre-trained on the task of language modeling, that is, predicting the next token in an unlabeled text corpus. GPT-4 can be accessed either using the chat completions API, or through an interactive chat interface. Notably, GPT-4 is closed-source, meaning the weights of the model have not been released, nor have the exact details of the architecture – including how many parameters the model has[1]. While earlier variants in the GPT family provide an API for fine-tuning, GPT-4 does not permit that as yet. Additionally, since the model requires passing datasets through a web-based API, which is subsequently stored temporarily on OpenAI's servers, there are concerns with using these models for sensitive data, such as our Sensor Immersion transcripts. The GPT-4 API is paid, and requests are charged based on the number of input and output tokens[2]. As a result, feeding in longer inputs results in a higher cost.

The Mistral [14] family of models are powerful open-source alternatives, developed by Mistral AI and released in November 2023. Crucially, the pre-trained weights of all model variants can be downloaded and used for local fine-tuning and exploration. Here we use Mistral-7B, a Transformer-based language model with 7 billion parameters.

Both models can perform in-context learning (generate predictions on unseen tasks or datasets using only context given in a prompt) [8]. This context can include just a task description, potentially framed as an instruction (zero-shot), or additionally include example demonstrations (few-shot).

Zero-shot prompting: In this setting, we create prompts instructing the model on what the task is, what form of input data will be provided, and the list of acceptable output labels. For each example in the test set, we concatenate the student utterance that needs to be classified to the prompt text, and query the model for a response. To obtain consistency, we use a temperature of zero. All prompts can be found in the appendix.

With GPT-4, following the official guidelines[3], the query is formatted as a sequence of two messages, each with a 'role' field to indicate the system or user, and a 'content' field for the text. The first message *prompts* the system with the task instruction. The second message contains each test example given by the user. Based on prompt engineering on the development set, we find that introducing a scenario in the beginning, such as "You are a teacher observing collaborative problem solving" is beneficial for more accurate predictions. We also instruct the model to produce single tokens corresponding to the labels in each dataset; any other responses are discarded. Following the advice of the Mistral developers, we format queries to include a short introduction to the scenario (which boosted results considerably), an opener for the utterance, the utterance in question, an opener for the label option, and a single label option. We pass each label from the label sets to the model in individual queries and we choose the label with the highest probability of appearing after the prompt as the prediction.

Few-shot prompting: Here, we concatenate k examples (shots) of *each class* to the prompts (formatted as described above) for each test instance. We choose them from the training set by randomly sampling k instances for each class. We repeat this process three times with different random seeds for sampling, thereby obtaining three different sets of k-shots. We then prompt the model in three runs, for each of the three k-shots, and report average scores across runs. While this doesn't exhaustively cover all possible k-shots from the training set, it smooths over random variation, giving a more fair estimate. We use the same k-shots for every test set query.

For GPT-4, we provide the k-shots by adding $2{*}k$ extra messages (utterance text under the user role and true labels under the system role) before the query. This informs the model that when a user provides a student utterance for classification, the desired *system* output is the true label. Similarly, for Mistral, we add $2{*}k$ extra messages to the sequence described above. This takes the form of the scenario followed by "#### Here are some examples: Student Utterance: {*example from train set*}. Student Action: {*true label from train set*}", and ending with the same query prompt as in the zero-shot scenario. We experiment with two values for k with Mistral: 5 and 20. We chose 5 and 20 per class to illustrate realistic low-data scenarios which are also reasonably proportionate to the size of our datasets. Similar values are also used in benchmarks for few-shot learning such as RAFT [1]. However, we only assess GPT-4 with

[1] Some estimates place this at over 1 trillion parameters.

[2] At the time of publication, the GPT-4 model costs $0.03 for each input token and $0.06 for each output token.

[3] `https://platform.openai.com/docs/guides/text-generation/chat-completions-api`

5-shot prompting due to the high cost of longer prompts.

4.2 Fine-Tuning

RoBERTa: The RoBERTa model [17] is an encoder-decoder model that is based on the Transformer [32] architecture. It is pre-trained with the objective of masked language modeling on a large training corpus. It learns rich, general purpose language representations language during pre-training, and can then be fine-tuned on task-specific data to learn a downstream task. This is a high-performing, popular model for classification tasks, both in education and more broadly in NLP. In our experiments, we use the pre-trained RoBERTa-base model consisting of 125M parameters, and add a sequence classification head that maps from the learned representations to each output label space. We perform full fine-tuning (i.e., updating all parameters with training data) using a CrossEntropy loss objective, and the AdamW optimizer. Each model is fine-tuned for up to 30 epochs with early stopping based on F1 score on the validation set. We use the implementation of the model available through the Huggingface Transformers library [35] and perform all training on a single Nvidia V100 GPU.

Mistral Embeddings: For fine-tuning the much larger mistral model, a resource-efficient approach is to extract text representations from the model and train a traditional machine learning classifier to predict the labels. One can obtain fixed-sized vector representations, or embeddings, for input text sequences by passing the input text through the LLM and retrieving the hidden states from one of the model's layers. These hidden states capture semantic and contextual information about the input, and the resulting embeddings serve as rich, contextualized representations of the input text, allowing a classifier to effectively learn decision boundaries.

We follow the implementation designed by the creators of Mistral [14]. Specifically, with an input utterance of N tokens, the Mistral model output is a list of N vectors (one vector per token) of dimension d = 4096 (the dimensionality of the model). The vectors are averaged along dimension 0 to get a single vector of size d that represents the full utterance. The embeddings were normalized to improve the performance and stability of the downstream classifier. In this scenario, we did not perform hyperparameter tuning on the classifier, but rather utilized the same Logistic Regression model (with parameters C = 1.0 and maximum iterations = 500) in each task and dataset.

Fine-Tuning in Data-Scarce Settings: To see how the task specific RoBERTa model compares to few-shot prompting LLMs, we artificially create a data-constrained setting by restricting the amount of data that RoBERTa has access to during fine-tuning. Using the same few-shot examples given to GPT-4, we fine-tune the pre-trained RoBERTa model on *only* the few-shot examples using the same hyperparameters for training as in Section 4.2. As in Section 4.1, we experiment with two choices of shots, 5 and 20, and for each, report average performance across three sets of shots. However, unlike few-shot *prompting*, we perform full fine-tuning with parameter updates in this setting.

5. RESULTS

Table 1 contains our main results, showing the performance of the in-context learning paradigm in comparison to task-specific fine-tuning across all datasets. As mentioned in Section 4.1, we do not use GPT-4 with the sensor immersion CPS dataset due to privacy and access restrictions. We report the F1-score of the positive class (e.g., presence of the CPS skill *negotiation*) for the binary classification tasks. For the multi-class datasets, specifically, CIMA and TalkMoves, we report the macro-averaged F1 score. We also include a random baseline for comparison, that randomly selects a valid label conditioned on the training set label distribution.

5.1 Zero-Shot Performance

In the zero-shot setting, the GPT-4 model outperforms the random baseline in 5 out of 6 tasks. It is particularly strong for multi-class classification: for student talk move prediction where there are five possible labels, zero-shot GPT-4 achieves an F1 of 0.42, whereas chance performance is 0.20. Similarly, for the CIMA dataset where there are four possible labels, zero-shot GPT-4 with an F1 of 0.49 vastly outperforms the random baseline of 0.29. On the binary classification tasks, however, results are mixed; in identifying student reasoning (NCTE), zero-shot GPT-4 is better than random prediction by 0.3 F1 points. For the CPS skill prediction tasks, it is at par with random on two tasks, and is weaker than the random baseline at predicting if utterances exhibit the skill *construction of shared knowledge*.

The Mistral model does not seem as adept at handling student data in the zero-shot setting as GPT-4. For the binary classification tasks, it is worse than random prediction on all tasks except predicting student reasoning. For both multi-class classification tasks, it is worse than random. We hypothesize that this stark difference is likely due to GPT-4's larger size and supervised fine-tuning on human responses leading to superior language understanding.

5.2 Few-Shot Performance

As outlined in Section 4.1, we only experiment with 5-shot learning on GPT-4 due to its high costs for longer inputs. The 5-shot examples are very helpful for detecting all CPS skills , especially for *construction of shared knowledge*, where performance improves from 0.26 F1 to 0.60 F1. Unfortunately, performance drops on the other datasets when the 5-shot demonstrations are provided. This could potentially be due to the model overfitting to the provided examples instead of using other cues to solve the task, such as its own pre-training knowledge.

When Mistral is prompted in the few-shot setting, we observe large improvements for 5-shot prompting on the binary classification tasks, surpassing random performance on the CPS-SI and NCTE datasets. However, performance is at par with zero-shot prompting for multi-class classification. Additionally, we see that increasing the number of shots to 20 does not help the Mistral model further, and in fact, hurts performance for most tasks. We leave it to future work to use improved methods for selecting the best possible demonstrations for each test example.

5.3 Fine-tuning Performance

Overall, the fine-tuned models that harness the entire training set are the most capable across the board. The RoBERTa

Table 1: F1 score performance of in-context learning vs task-specific fine-tuning. Positive class F1 for binary classification, macro-averaged F1 for multi-class classification. (CONST: ***constructing shared knowledge*****, NEG:** ***negotiation / coordination*****, MAINT:** ***maintaining team function*****, SI:** ***Sensor Immersion*****)**

Paradigm	Model	CPS Minecraft			CPS SI			Talk	CIMA	NCTE
		CONST	NEG	MAINT	CONST	NEG	MAINT	Move		
Baseline	Random	0.34	0.15	0.10	0.23	0.16	0.07	0.20	0.29	0.19
In-Context Learning	0-Shot–GPT-4	0.26	0.17	0.11	-	-	-	0.42	0.49	0.49
	5-Shot–GPT-4	0.60	0.27	0.21	-	-	-	0.27	0.45	0.38
	0-Shot–Mistral	0.02	0.09	0.01	0.07	0.15	0.15	0.08	0.11	0.23
	5-Shot–Mistral	0.25	0.10	0.12	0.33	0.21	0.15	0.09	0.11	0.24
	20-Shot–Mistral	0.00	0.02	0.00	0.01	0.00	0.00	0.11	0.20	0.30
Fine-tune (Train Set)	RoBERTa	**0.71**	**0.53**	**0.33**	**0.61**	**0.51**	**0.32**	**0.67**	**0.63**	**0.67**
	Mistral Embeddings	0.59	0.32	0.26	0.59	0.44	0.13	0.53	0.62	0.63
Fine-tune (Subsets of Train Set)	5-Shot–RoBERTa	0.66	0.31	0.24	0.40	0.26	0.10	0.35	0.48	0.40
	20-Shot–RoBERTa	0.66	0.32	0.25	0.50	0.36	0.15	0.42	0.55	0.54

model fine-tuned on all training samples is the best out of all our models and tasks. It also vastly outperforms the random baseline on every dataset. The Mistral Embeddings model is not as strong, although it outperforms the prompt-based models on all datasets. We hypothesize that the large size of the Mistral model may affect the model's generalizability, especially given that our datasets are small. However, in comparison to its few-shot performance, the Mistral model shows a remarkable improvement when trained with task-specific data. Our results are also in line with other observations that the RoBERTa model can be a better choice than very large language models in classifying short sequences [13].

We also find that the RoBERTa model is surprisingly effective in the few-shot training setting, even with only 5 and 20 training examples per class. In the 5-shot setting, it outperforms all prompting methods on all CPS tasks except for predicting the skill of *maintaining team function*. On the multi-class datasets, and the NCTE dataset, it outperforms the random baseline but is not as good as zero-shot GPT-4. When the number of training examples is increased to 20 per class, the RoBERTa model performs at par or better than all prompt-based models on all datasets.

6. DISCUSSION

In this paper, we set out to investigate whether the high performance of off-the-shelf LLMs on NLP tasks translates to challenging qualitative coding tasks in education. We find that advanced models like GPT-4 do outperform random-chance baselines in the zero-shot setting, indicating that their pre-training process has imbued them with some understanding about contexts such as classroom dialog. However, the extent of this performance is highly task-specific. We see that GPT-4 does well without any examples on a task like determining if an utterance exhibits reasoning (NCTE), where the objective can be reasonably understood through a simple task description, and the solution could make use of cue words like 'because'. However, the models struggle particularly on complex, theoretically motivated tasks like CPS until examples are provided, after which performance sharply improves. We also note that GPT-4 and Mistral exhibit stark differences in performance, highlighting the importance of model choice in choosing between prompting vs task-specific fine-tuning. One important consideration is that the design of the prompt may have a big influence on zero and few-shot performance. While we follow best practices recommended by developers, and experimented with framings such as scenarios and role-playing, there are other strategies for prompt engineering that could prove useful, such as feeding an entire codebook in a prompt. Given that this may come with the trade-off of high costs, we do not carry out extensive prompt engineering at this stage.

Next, in reflecting on whether LLMs obviate the need for collecting large-scale annotated training datasets, we show that these models cannot yet replace the traditional pre-training–fine-tuning paradigm, particularly for qualitative coding. As a result, we argue that training data is still highly valuable, particularly for nuanced, subjective tasks that benefit from numerous examples, such as student talk move or CPS skill prediction. In situations where access to data is severely constrained, models like GPT-4 show potential, strongly outperforming random guessing. However, prompting with few-shot examples may not necessarily be an improvement over the zero-shot setting: the choice and quality of shots is crucial and may lead to variability. If cost is also a factor in the data-constrained setting, an alternative to few-shot prompting through expensive APIs can be few-shot fine-tuning with models like RoBERTa, which achieve surprisingly high performance on coding tasks with very limited training data.

Finally, when assessing which automated qualitative coding models may be best suited for designing systems for analyzing classroom conversations, we find that training small, accessible, and inexpensive models like RoBERTa with high-quality data may still lead to better performance than using off-the-shelf LLMs. While these models are still outstanding at text generation tasks, and therefore highly relevant for educational applications, we conclude that they show limited promise as of now for classification-oriented qualitative coding tasks in education.

7. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their feedback and suggestions. This research was supported by the NSF National AI Institute for Student-AI Teaming (iSAT) under grant DRL 2019805 and under grant DRL 1920510. The opinions expressed are those of the authors and do not represent views of the NSF.

8. REFERENCES

[1] N. Alex, E. Lifland, L. Tunstall, A. Thakur, P. Maham, C. J. Riedel, E. Hine, C. Ashurst, P. Sedille, A. Carlier, et al. Raft: A real-world few-shot text classification benchmark. *arXiv preprint arXiv:2109.14076*, 2021.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] C. Chandler, T. Breideband, J. G. Reitman, M. Chitwood, J. B. Bush, A. Howard, S. Leonhart, P. W. Foltz, W. R. Penuel, and S. K. D'Mello. Computational modeling of collaborative discourse to enable feedback and reflection in middle school classrooms. In *Proceedings of the 14th Learning Analytics and Knowledge Conference (LAK'24), March 18–22, 2024*, page 11 pages, Kyoto, Japan, 2024. ACM, New York, NY, USA.

[4] S.-C. Dai, A. Xiong, and L.-W. Ku. LLM-in-the-loop: Leveraging large language model for thematic analysis. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9993–10001, Singapore, Dec. 2023. Association for Computational Linguistics.

[5] W. Dai, J. Lin, H. Jin, T. Li, Y.-S. Tsai, D. Gašević, and G. Chen. Can large language models provide feedback to students? a case study on chatgpt. In *2023 IEEE International Conference on Advanced Learning Technologies (ICALT)*, pages 323–325. IEEE, 2023.

[6] D. Demszky and H. Hill. The NCTE transcripts: A dataset of elementary math classroom transcripts. In E. Kochmar, J. Burstein, A. Horbach, R. Laarmann-Quante, N. Madnani, A. Tack, V. Yaneva, Z. Yuan, and T. Zesch, editors, *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 528–538, Toronto, Canada, July 2023. Association for Computational Linguistics.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

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

[9] S. E. Finch, E. S. Paek, and J. D. Choi. Leveraging large language models for automated dialogue analysis. In S. Stoyanchev, S. Joty, D. Schlangen, O. Dusek, C. Kennington, and M. Alikhani, editors, *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 202–215, Prague, Czechia, Sept. 2023. Association for Computational Linguistics.

[10] A. Ganesh, M. A. Chang, R. Dickler, M. Regan, J. Cai, K. Wright-Bettner, J. Pustejovsky, J. Martin, J. Flanigan, M. Palmer, et al. Navigating wanderland: Highlighting off-task discussions in classrooms. In *International Conference on Artificial Intelligence in Education*, pages 727–732. Springer, 2023.

[11] K. Handa, M. Clapper, J. Boyle, R. Wang, D. Yang, D. Yeager, and D. Demszky. "mistakes help us grow": Facilitating and evaluating growth mindset supportive language in classrooms. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8877–8897, Singapore, Dec. 2023. Association for Computational Linguistics.

[12] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[13] Huggingface. Comparing the performance of llms: A deep dive into roberta, llama 2, and mistral for disaster tweets analysis with lora, November 2023.

[14] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b, 2023.

[15] A. Kupor, C. Morgan, and D. Demszky. Measuring five accountable talk moves to improve instruction at scale, 2023.

[16] 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 *International Conference on Artificial Intelligence in Education*, pages 114–125. Springer, 2023.

[17] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[18] J. Meyer, T. Jansen, R. Schiller, L. W. Liebenow, M. Steinbach, A. Horbach, and J. Fleckenstein. Using llms to bring evidence-based feedback into the classroom: Ai-generated feedback increases secondary students' text revision, motivation, and positive emotions. *Computers and Education: Artificial Intelligence*, 6:100199, 2024.

[19] S. Moore, R. Tong, A. Singh, Z. Liu, X. Hu, Y. Lu, J. Liang, C. Cao, H. Khosravi, P. Denny, et al. Empowering education with llms-the next-gen interface and content generation. In *International Conference on Artificial Intelligence in Education*, pages 32–37. Springer, 2023.

[20] B. Naismith, P. Mulcaire, and J. Burstein. Automated evaluation of written discourse coherence using GPT-4. In E. Kochmar, J. Burstein, A. Horbach,

R. Laarmann-Quante, N. Madnani, A. Tack, V. Yaneva, Z. Yuan, and T. Zesch, editors, *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 394–403, Toronto, Canada, July 2023. Association for Computational Linguistics.

[21] H. Nori, Y. T. Lee, S. Zhang, D. Carignan, R. Edgar, N. Fusi, N. King, J. Larson, Y. Li, W. Liu, R. Luo, S. M. McKinney, R. O. Ness, H. Poon, T. Qin, N. Usuyama, C. White, and E. Horvitz. Can generalist foundation models outcompete special-purpose tuning? case study in medicine, 2023.

[22] OpenAI, J. Achiam, and S. A. et al. Gpt-4 technical report, 2023.

[23] P. Organisciak, S. Acar, D. Dumas, and K. Berthiaume. Beyond semantic distance: automated scoring of divergent thinking greatly improves with large language models. *Thinking Skills and Creativity*, page 101356, 2023.

[24] N. Pangakis, S. Wolken, and N. Fasching. Automated annotation with generative ai requires validation. *arXiv preprint arXiv:2306.00176*, 2023.

[25] Z. A. Pardos and S. Bhandari. Learning gain differences between chatgpt and human tutor generated algebra hints. *CoRR*, abs/2302.06871, 2023.

[26] W. A. Sahlman, A. M. Ciechanover, and E. Grandjean. Khanmigo: Revolutionizing learning with genai, November 2023.

[27] M. Sclar, Y. Choi, Y. Tsvetkov, and A. Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting, 2023.

[28] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

[29] A. E. Stewart, M. J. Amon, N. D. Duran, and S. K. D'Mello. Beyond team makeup: Diversity in teams predicts valued outcomes in computer-mediated collaborations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.

[30] A. Suresh, J. Jacobs, C. Harty, M. Perkoff, J. H. Martin, and T. Sumner. The TalkMoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4654–4662, Marseille, France, June 2022. European Language Resources Association.

[31] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[33] R. Wang and D. Demszky. Is ChatGPT a good teacher coach? measuring zero-shot performance for scoring and providing actionable insights on classroom instruction. In E. Kochmar, J. Burstein, A. Horbach, R. Laarmann-Quante, N. Madnani, A. Tack, V. Yaneva, Z. Yuan, and T. Zesch, editors, *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 626–667, Toronto, Canada, July 2023. Association for Computational Linguistics.

[34] R. Wang, P. Wirawarn, N. Goodman, and D. Demszky. SIGHT: A large annotated dataset on student insights gathered from higher education transcripts. In E. Kochmar, J. Burstein, A. Horbach, R. Laarmann-Quante, N. Madnani, A. Tack, V. Yaneva, Z. Yuan, and T. Zesch, editors, *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 315–351, Toronto, Canada, July 2023. Association for Computational Linguistics.

[35] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

[36] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, and Q. Yang. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2023.

[37] C. Ziems, W. Held, O. Shaikh, J. Chen, Z. Zhang, and D. Yang. Can Large Language Models Transform Computational Social Science? *Computational Linguistics*, pages 1–55, 02 2024.

9. APPENDIX

Table 2: Prompts utilized for GPT-4 and Mistral zero-shot learning. For binary scenarios, Mistral is prompted with "Yes" or "No" as response options ({*Yes/No*}), while multiclass scenarios involve a generic prompt for a label and the iterative testing of each label option ({*label*})

Task	**GPT-4 Prompts**	**Mistral Prompts**

NCTE	{"role": "system", "content": "You're observing students collaboratively solving math problems. Determine if their response displays reasoning skills, return 1 if so and 0 otherwise."},
	{"role": "user", "content": "Utterance: {*utterance*}"
		Scenario: You're observing students collaboratively solving math problems.
		Determine if their response displays reasoning skills.
		Student Utterance: {*utterance*}
		Contains Student Reasoning: {*Yes/No*}

CPS-COMM	{"role": "system", "content": "You're observing students working collaboratively. Determine if the utterance displays the collaborative problem solving skill of construction of shared knowledge. Return 1 if the skill is shown, and 0 otherwise."}, {"role": "user", "content": "Utterance: {*utterance*}"	Scenario: You're observing students working collaboratively. Determine if the utterance exhibits the construction of shared knowledge. Student Utterance: {*utterance*} Exhibits the construction of shared knowledge: {*Yes/No*}

CPS-NEG	{"role": "system", "content": "You're observing students working collaboratively. Determine if the utterance displays the collaborative problem solving skill of negotiation or coordination. Return 1 if the skill is shown, and 0 otherwise."},
	{"role": "user", "content": "Utterance: {*utterance*}"
		Scenario: You're observing students working collaboratively. Determine if the utterance exhibits the negotiation / coordination.
		Student Utterance: {*utterance*}
		Exhibits negotiation / coordination: {*Yes/No*}

CPS-NEG	{"role": "system", "content": "You're observing students working collaboratively. Determine if the utterance displays the collaborative problem solving skill of maintaining team function. Return 1 if the skill is shown, and 0 otherwise."}, {"role": "user", "content": "Utterance: {*utterance*}"	Scenario: You're observing students working collaboratively. Determine if the utterance exhibits the maintaining team function. Student Utterance: {*utterance*} Exhibits maintaining team function: {*Yes/No*}

CIMA	{"role": "system", "content": "You're observing a student learning Italian prepositions.
	Classify their response into one out of 4 categories: [Guess, Question, Affirmation, Other].
	Only return the label corresponding to one of the four categories.}
	{"role": "user", "content": "Utterance: {*utterance*}"
		Scenario: You're observing a student learning Italian prepositions.
		Student Utterance: {*utterance*}
		Student Action: {*label*}

Talk Move	{"role": "system", "content": "You're observing students in a math classroom. Determine what talk move they are using by looking at both the student utterance and the context.
	There are 5 talk moves indexed from 0 to 4:
	0=None, 1=Relating to Another Student
	2=Ask for more info, 3=Make a claim,
	4=Provide evidence and reasoning. Return the index of the correct talk move."},
	{"role": "user", "content": "Utterance: {*utterance + prev*}"
		Scenario: You're observing students working with a tutor on math problems.
		Tutor Utterance: {*prev. tutor utterance*}
		Student Utterance: {*student utterance*}
		Student Action: {*label*}

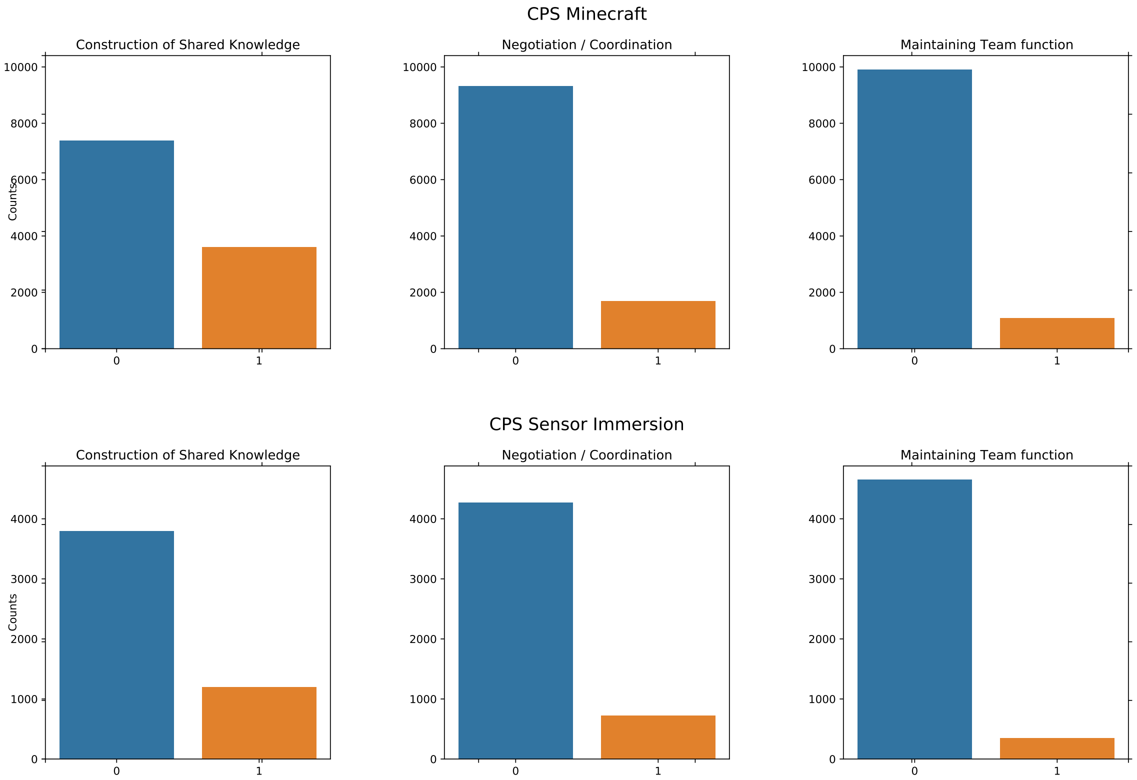

Figure 1: Distribution of labels in both CPS skills datasets

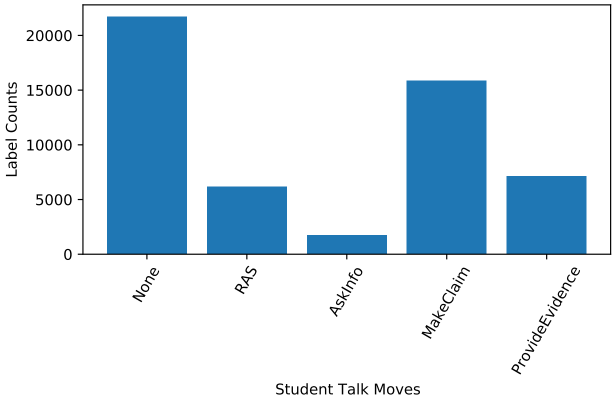

Figure 2: Distribution of labels in the TalkMoves dataset

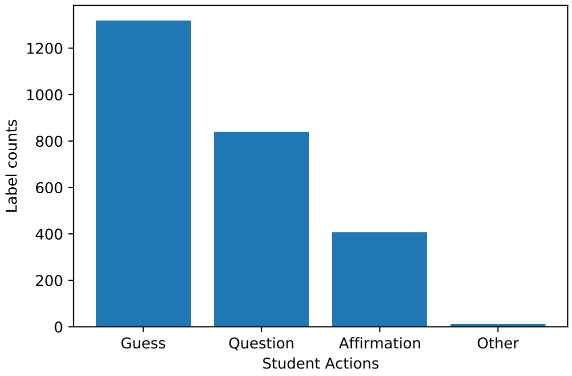

Figure 3: Distribution of labels in the CIMA corpus