Using LLMs to Simulate Students' Responses to Exam Questions

Luca Benedetto[♥], **Giovanni Aradelli**[♣], **Antonia Donvito**[♣], **Alberto Lucchetti**[♣], **Andrea Cappelli**[♣], **Paula Buttery**[♥]
[♥]ALTA Institute, University of Cambridge, UK
[♣]QA Ltd., Mendrisio, Switzerland
{name.surname}@cl.cam.ac.uk
{name.surname}@qa.com

Abstract

Previous research leveraged Large Language Models (LLMs) in numerous ways in the educational domain. Here, we show that they can be used to answer exam questions simulating students of different skill levels and share a prompt, engineered for GPT-3.5, that enables the simulation of varying student skill levels on questions from different educational domains. We evaluate the proposed prompt on three publicly available datasets (one from science exams and two from English reading comprehension exams) and three LLMs (two versions of GPT-3.5 and one of GPT-4), and show that it is robust to different educational domains and capable of generalising to data unseen during the prompt engineering phase. We also show that, being engineered for a specific version of GPT-3.5, the prompt does not generalise well to different LLMs, stressing the need for prompt engineering for each model in practical applications. Lastly, we find that there is not a direct correlation between the quality of the rationales obtained with chain-of-thought prompting and the accuracy in the student simulation task.

1 Introduction

Large Language Models (LLMs) currently represent the state of the art in text generation, with some capable of generating human-like texts, such as GPT-4 (OpenAI, 2023), Llama (Touvron et al., 2023; Meta, 2024), and Gemma (Gemma Team and DeepMind, 2024). In this work we focus on the educational domain, which can massively benefit from LLMs (Jeon and Lee, 2023; Kasneci et al., 2023; Caines et al., 2023). Specifically, we study whether it is possible to leverage LLMs to simulate the response patterns of students of different skill levels to exam questions. Previous research tried to simulate the responses of human participants to surveys with LLMs (Dillion et al., 2023; Argyle et al., 2023; Demszky et al., 2023; Aher et al., 2023), but nothing similar has been done for simulating students answering exam questions. There have been some concerns about the fairness of using LLMs instead of (or in addition to) human survey participants (Harding et al., 2023; Crockett and Messeri, 2023), and we agree that this is an important aspect to consider in the educational domain, as well. However, we believe that it might be less of an issue with respect to general-domain surveys, due to the factual nature of learning content and exam questions, which are built to evaluate domain knowledge and to minimise the effects that the wording has on the students' outcomes (Yaneva et al., 2019). In this work, we aim at answering the following Research Questions. **RQ1**: can LLMs be prompted to answer Multiple Choice Questions (MCQs) while role-playing as (i.e., simulating) learners of different skill levels, and does this generalise to unseen data?[1] **RQ2**: How do these findings compare across different models?

We work primarily on GPT-3.5[2] and three publicly available datasets of science MCQs (*ARC*) and English reading comprehension MCQs (*RACE* and *CUP&A*), and engineer a prompt (referred to as "Reference Prompt" or `RP`) that leads the LLM to answer exam questions with different levels of accuracy, thus representing students of different skill levels. Also, we observe a small but positive correlation between the difficulty obtained from virtual pretesting with LLMs and the difficulty from pretesting with human learners. Although the prompt was engineered using only GPT-3.5 and one dataset, this behaviour proved generalisable to new questions (also from different educational domains), but not to other LLMs, thus stressing the need for prompt engineering for each model. Lastly, we find that there is not a direct correlation between the quality of the rationales obtained with chain-of-thought prompting and the accuracy of the models in the student simulation task.

[1] *Unseen* indicates data not used for prompt engineering.
[2] We use *gpt-3.5-turbo-0613*, except where explicitly said.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 11351–11368
November 12-16, 2024

The code, prompts, and LLM outputs are publicly available at github.com/lucabenedetto/LLM-student-simulations.

2 Methodology

We proceed in three steps: we i) perform prompt engineering to search for the "best" prompt using one LLM and one dataset, ii) evaluate its generalisation capabilities to unseen data and to other LLMs, and iii) perform some additional analyses of the models' responses.

2.1 Prompt engineering

We perform prompt engineering on GPT-3.5 and a *dev* set subsampled from the *ARC* dataset (questions from science exams), considering only zero-shot prompts and *temperature=0*.[3] The LLM is prompted to perform MCQ Answering (MCQA) simulating students of different skill levels and, with each prompt, we ask the model to simulate one student level and answer one question. Thus, in our setup the LLM is shown only one question at a time, without having a view of the whole exam or information about its previous responses. Similarly, the LLM is asked to simulate one student at a time, not to provide in a single response the answers of students of different levels. From this initial exploration we develop the "*reference prompt*" (RP), which is the one that leads to the best simulation of students' response patterns, according to the metrics defined in 3.2. Specifically, we are looking for i) increasing MCQA accuracy for increasing simulated levels, and ii) lower accuracy on more difficult questions. RP is shown in Table 1.

2.2 Analysis of generalisation capabilities

Generalisation to unseen data. We study the generalisation capabilities of RP as follows. We evaluate it i) on a separate *test* set from *ARC*, and ii) on *RACE* and *CUP&A*, which contain English reading comprehension questions.[4] This approach might penalise the LLM, as the prompt was not engineered on these datasets, but we argue that it is the most appropriate way to study the generalisation capabilities of the proposed method, as it is the standard methodology of splitting the dataset in *dev* and *test* sets.

[3]To reduce the variance of the LLM output.

[4]As shown in Table 1, RP is actually slightly changed, adding the text of the reading passage and swapping *a science exam* with *an English reading comprehension exam*, to reflect the different nature of these datasets.

Generalisation to other LLMs. Prompt engineering was performed on *gpt-3.5-turbo-0613* but we also experiment on using RP on two different LLMs, *gpt-3.5-turbo-1106* and *gpt-4-1106-preview*, to see whether the behaviour generalises and is consistent across models.

2.3 Additional analyses

***Question level* and *answer explanation* fields.** As shown in Table 1, RP asks the LLM not only to answer the question as a student of a specific level, but also to i) assign a difficulty level to the question (*question level* in the output JSON) and ii) explain its rationale (*answer explanation*). Although we added these two fields to RP because they proved helpful in reaching the desired simulation capabilities, by leveraging chain-of-thought prompting (Wei et al., 2022), we evaluate them to see whether they can provide useful insights. Specifically, we compare the model-assigned difficulty with the reference difficulty levels available in the datasets, and perform a quantitative and qualitative analysis of the explanations to study whether there are meaningful differences between simulated levels and their educational validity.

Experiments on different educational scales. RP simulates students on an abstract knowledge scale from *one* to *five*, but we also study the effects of using different scales. Specifically, we consider: i) *exam marks* (*A, ..., F*) and ii) a non-standardised scale (*beginner, intermediate, advanced*).

3 Experimental Setup

3.1 Experimental datasets

We experiment with three public datasets.

ARC, AI2's Reasoning Challenge dataset (Clark et al., 2018), is a MCQA dataset of questions from science exams. Each question is assigned a *grade* (from 3 to 9), which indicates the school grade that the question was built for. Although this is not a direct indication of question difficulty, questions with higher grades are meant for more advanced learners, and the *grade* has been used as a proxy for question difficulty in previous research (Benedetto, 2023). We work on a subsampled portion of the dataset: we use 350 questions as *dev set* and other 350 as *test set*. Both sets are sampled from the original *test* split with stratified sampling in order to have in both groups 50 questions for each grade.

RACE is a MCQA dataset of questions from English reading comprehension exams. We work on

Table 1: Reference prompt RP for the *ARC* dataset, the variable {X} in the system message is substituted with *one*, *two*, ..., *five* to indicate one of five student levels. The reference prompt for *RACE* and *CUP&A* is the same, except two changes done to account for the questions of different type: i) *a science exam* is swapped with *an English reading comprehension exam* and ii) we add *Reading passage: "{passage}"* to the user prompt (before *Question*).

SYSTEM:
You will be shown a multiple choice question from a *science exam*, and the questions in the exam have difficulty levels on a scale from one (very easy) to five (very difficult). You must assign a difficulty level to the given multiple choice question, and select the answer choice that a student of level {X} would pick.
Provide only a JSON file with the following structure: {"question level": "difficulty level of the question", "answer explanation": "the list of steps that the students of level {X} would follow to select the answer, including the misconceptions that might cause them to make mistakes", "index": "integer index of the answer chosen by a student of level {X}"}
USER:
Question: "{question}"
Options: "{answer options}"

the version obtained by merging the original RACE (Lai et al., 2017) with RACE-c (Liang et al., 2019). Each question in the dataset is assigned one of three *levels* (*middle*, *high*, *college*), which indicates the school level of the target students. Similarly to *ARC*, although this is not a direct indication of question difficulty, it has been used as a proxy for it in previous research, *middle* being the lowest difficulty and *college* the highest, e.g., by Loginova et al. (2021). We work on a reduced set of 150 questions, obtained with stratified sampling from the *test* split, keeping 50 questions per level.

CUP&A[5] (Mullooly et al., 2023), is a MCQA dataset of questions from English reading comprehension exams. It contains questions aimed at students of different CEFR levels (from B1 to C2). We work on a stratified version built by sampling 50 questions for each CEFR level (200 questions in total). Differently from *ARC* and *RACE*, it provides for all the questions an indication of the actual question difficulty, as obtained from pretesting with real learners. This can be compared with the difficulty obtained from virtual pretesting with LLMs.

3.2 Metrics for prompt engineering

Evaluating whether the LLMs are capable of simulating students is not straightforward, especially considering the information that is available in publicly available datasets. Indeed, the ideal evaluation would be to compare the response pattern of the LLMs with the response patterns of human learners, but the latter is unavailable in the three datasets we experiment on.[6] As an alternative, we evaluate each prompt as follows.

i) We study the MCQA accuracy of the LLMs when simulating students of different levels; ideally, we want a monotonically increasing accuracy curve (i.e., higher role-played levels are more accurate).[7] Specifically, we devise an evaluation metric to quantitatively compare the accuracy plots obtained with different prompts. The metric combines the correlation with the ideal accuracy curve and penalizes non-monotonic behavior in the accuracy sequence. Let $\mathbf{L} = (a_1, a_2, \cdots, a_5)$ be the list of accuracy scores obtained when using one prompt to simulate levels *(one, two, ..., five)* and $\mathbf{I} = (0.0, 0.25, 0.5, 0.75, 1.0)$ the ideal accuracy curve. We refer with $\rho_{\mathbf{L},\mathbf{I}}$ to the Pearson's correlation between the two accuracy curves. The Penalty for Non-Monotonicity (P) is calculated as: $\sum_{i=1}^{4} \sqrt{|a_{i+1} - a_i|} \cdot \mathbb{I}(a_{i+1} < a_i)$, where $\mathbb{I}(a_{i+1} < a_i)$ is an indicator function that is 1 when $a_{i+1} < a_i$ and 0 otherwise. Finally, the metric (M) is the difference between the correlation score and the penalty for non-monotonicity: $M = \rho_{\mathbf{L},\mathbf{I}} - P$.

ii) We also analyse the MCQA accuracy of different simulated levels on questions of different difficulty: given a simulated level, the MCQA accuracy should be lower on more difficult questions.

4 Results and Analysis

4.1 Analysis of the reference prompt

Our first step consists in engineering the *reference prompt* (RP), by iterating over a number of different prompts and evaluating them with GPT-3.5 on the *dev* set of *ARC*. Figure 1 shows how the MCQA accuracy of GPT-3.5 changes depending on the role-played level for five different prompts; all the prompts shown here use non-standardised

[5]The *Cambridge MCQs Reading Dataset* from Cambridge University Press & Assessment.

[6]To the best of our knowledge, there are no public datasets providing both the texts of MCQs and the students' responses.

[7]In our experimental setting, each simulated learner answer all the questions of the exam, meaning that the accuracy corresponds to the estimated knowledge level according to testing theories such as IRT (Hambleton et al., 1991).

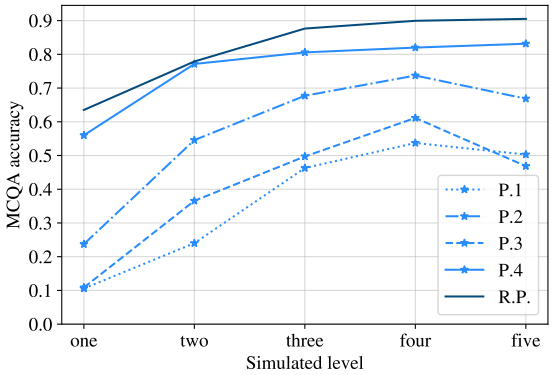


Figure 1: Comparison of the MCQA accuracy of GPT-3.5 on the *dev* split of *ARC*, when prompted with different prompts to simulate students of different levels.

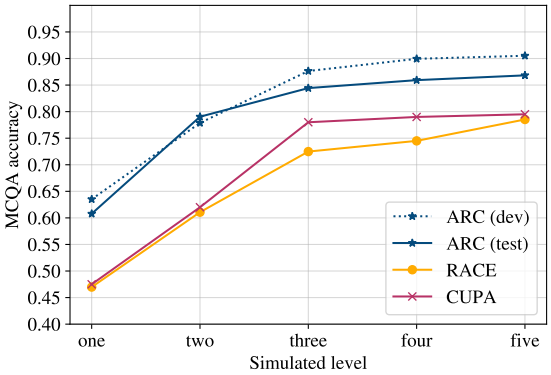


Figure 2: MCQA accuracy of different simulated levels obtained with the reference prompts and GPT-3.5 on the *test* split of *ARC* and on the *RACE* and *CUP&A* datasets.

students' levels from *one* to *five*. The reference prompt (`RP`), which is the one we selected as best performing on the *dev* set of *ARC*, leads to the highest score according to the metric defined in Section 3.2: $M_{\text{RP}} = 0.91$. Compared to `RP`, i) prompt `P1` ($M_{\text{P1}} = 0.73$) adds a description about the meaning of students' levels; ii) prompt `P2` ($M_{\text{P2}} = 0.57$) removes the *answer explanation* field and adds a field for the text of the chosen answer; iii) prompt `P3` ($M_{\text{P3}} = 0.43$) adds a description of each level to prompt `P2`; and iv) prompt `P4` ($M_{\text{P4}} = 0.83$), the most similar to the reference prompt, renames the field *answer explanation* into *motivation*. The complete prompts are shown in Appendix A.

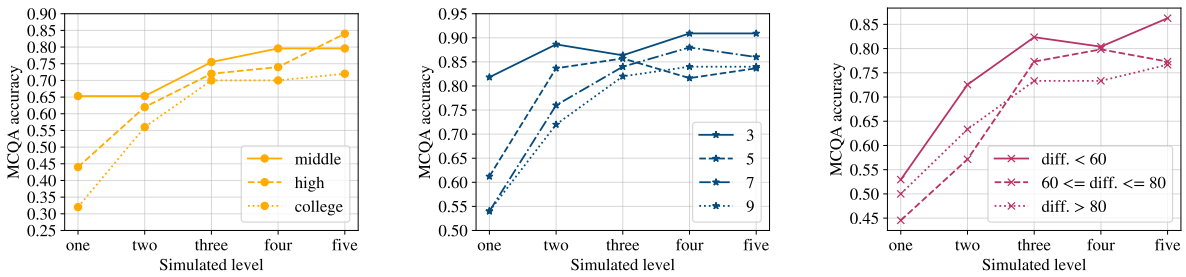
A common issue is to have the highest MCQA accuracy for intermediate (simulated) levels – shown in the figure by prompts `P1`, `P2`, and `P3` – and we observed this was often triggered by minor changes. Prompt `P4` is close to the desired behaviour (the trend is monotonic), but it shows a significant step in accuracy between simulated levels *one* and *two*, and then the accuracy almost reaches a *plateau*, which is undesirable, and indeed it leads to a lower score with respect to `RP`. The difference between `P4` and `RP` is only a renamed field in the output JSON required from the model, showing that even minor differences in the prompt can lead to relevant differences in the output. To have a reference, we also prompted GPT-3.5 to just answer the exam questions (without simulating a specific level) and obtained an accuracy of 0.92, slightly better than the highest simulated level.

4.2 Generalisation to unseen data

Figure 2 shows the behaviour of GPT-3.5 with `RP` on the *ARC dev* and *test* sets and the two reading comprehension datasets, *RACE* and *CUP&A*. The figure shows that the behaviour is qualitatively similar, with a monotonically increasing accuracy for increasing levels. As could be expected, it is slightly worse on the *ARC test* set than on the *dev* set: indeed, there is a smaller difference between the accuracy of the three highest simulated levels and this is captured by the metric M, which is 0.86 for *ARC test* (it was 0.91 for *dev*). Considering the other datasets, *CUP&A* (which shows an almost ideal trend for the first three levels but then reaches a *plateau*) scores 0.90, and *RACE* 0.94. These similar trends suggest that the behaviour obtained with `RP`[8] and GPT-3.5 is capable of generalising fairly well to previously unseen data, also coming from different educational domains. It is worth noting that the accuracy is generally higher on *ARC* than on the other two datasets; this might be because it is inherently easier for GPT (indeed, "just" GPT-3.5 has an accuracy of 0.86 on *ARC* test, 0.78 on *RACE* and 0.77 on *CUP&A*) or that we performed prompt engineering on it.

Focusing on the second metric defined in Section 3.2, we plot in Figure 3 the MCQA accuracy of different simulated levels on questions of different difficulty, separately for the three datasets. The figures show that the trend of increasing MCQA accuracy for increasing simulated levels is visible across question levels and across datasets. Focus-

[8]Please note that the reference prompt for *RACE* and *CUP&A* is slightly modified, as described in Table1.



(a) *RACE*, questions of different *levels*. (b) *ARC*, questions of different *grades*. (c) *CUP&A*, different *difficulty*.

Figure 3: Evaluation of the MCQA accuracy of GPT-3.5 on the three datasets when simulating students of different levels, separately on questions of different difficulty levels (difficulty definition is different in the three datasets).

ing on *RACE* (Figure 3a), if we look at the accuracy of a simulated level on questions of increasing levels, we can see that it consistently decreases, with the only exception of student level *five* on *high* questions. The same analysis is shown for *ARC* in Figure 3b.[9] The results are not as clean as on *RACE*: indeed, although we can see a general trend of increasing accuracy for increasing role-played levels, the trend is monotonic only for grade 9; grades 3 and 7 have one "drop" that affects monotonicity (level *three* and *five*, respectively), while grade 5 has several oscillations. Even though it is not always true that the same role-played level has lower accuracy on questions of higher grades, this trend is mostly visible for all grades, except grade 5 which seems to be the most problematic. This might also be due to the specific types of questions in *ARC*: indeed, even though most of the questions are *knowledge questions* for which it makes sense to define the difficulty, we observed that some do not necessarily get more difficult for higher grades (e.g., questions about safety equipment in the lab). Finally, Figure 3c shows the same results for *CUP&A*; in this case, the difficulty is a continuous value instead of a discrete one, hence we group questions in three difficulty bins. Again, the trend of monotonic increase in MCQA accuracy is visible across difficulty levels (with the only exceptions of level *four* for the easier questions and level *five* for the medium difficulty questions), and a given simulated level has lower accuracy on more difficult questions, with the only exceptions of simulated levels *one* and *two* on mid-level questions.

4.3 Virtual pretesting with role-playing LLMs

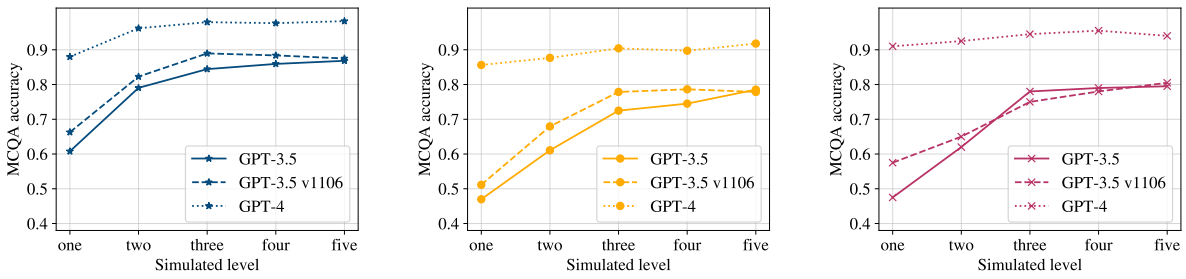
CUP&A provides for each question a quantitative measurement of difficulty obtained from pretesting with human learners. This enables us to evaluate the simulation capabilities of the LLM by performing virtual pretesting and comparing the difficulty obtained from it with the reference value, ideally looking for a perfect correlation.[10] In our setup, the difficulty from virtual pretesting is defined as the fraction of (simulated) students that answer the question wrongly. We observe a positive correlation coefficient between the two variables (difficulty from virtual and "real" pretesting) of 0.13 ($pvalue = 0.06$), while a random baseline leads to a correlation coefficient of -0.03 ($pvalue = 0.62$). This correlation might seem low but, to put it in context, we also performed an Item Response Theory (IRT) simulation (Hambleton et al., 1991). This consists in simulating the responses of five "mock" students of prescribed skill levels to the questions of known difficulty, and doing pretesting with such responses. We consider students' skills equally spaced in the skill range, which is an ideal scenario (almost) never observed in practice and can be seen as an upper bound. This simulation, whose details are described in Appendix B, led to a correlation of 0.43 ($pvalue = 10e^{-10}$). This suggests that, although the correlation observed with the LLMs is quite low, it is promising because a five-student pretesting scenario is particularly challenging, especially considering that the LLM is not given any anchoring item to match the simulated skill levels to the difficulty of the items in the exam.

4.4 Generalisation to other LLMs

To evaluate the generalisation capabilities of the reference prompt to other LLMs, we experiment with

[9] We show only the odd *grades* to improve readability.

[10] A short premise: at this stage, we are performing virtual pretesting with only five simulated students (GPT-3.5 role-playing as five students of different levels), which would be quite a small pretesting sample even with human learners.



(a) *ARC*. (b) *RACE*. (c) *CUP&A*.

Figure 4: Comparison between the behaviour of *gpt-3.5-turbo-0613* (full lines), *gpt-3.5-turbo-1106* (dashed lines), and *gpt-4-1106-preview* (dotted lines) on the three datasets, when prompted with the reference prompts.

a different version of GPT-3.5 (*gpt-3.5-turbo-1106*) and GPT-4 (*gpt-4-1106-preview*), and compare the behaviours obtained with them with *gpt-3.5-turbo-0613*, which is the version used for prompt engineering and all the other experiments. The results are shown separately for each dataset in Figure 4.

The updated version of GPT-3.5 shows a similar behaviour, but there are significant differences. For both *ARC* and *RACE* the behaviour is arguably worse, since the highest MCQA accuracy is not obtained for level *five* but instead for level *three* and *four* respectively, and the model reaches a plateau at level *three* for both datasets. This is also shown by the score obtained with the evaluation metric, which is 0.63 for *ARC* and 0.77 for *RACE* (it was 0.86 and 0.95 with GPT-3.5, respectively). The behaviour on *CUP&A* is different, though: the newer version performs actually better since it does not reach a plateau, and indeed gets a higher score of 0.96 (it was 0.90). It is also worth remarking that, in almost all cases, the newer *gpt-3.5-turbo-1106* leads to higher MCQA accuracy, and a narrower range of skill levels for virtual pretesting. These results suggest that prompts engineered for a specific version of GPT-3.5 should only be used on that specific version, as they might work differently when used on different versions.

On the other hand, the behaviour with GPT-4 is clearly worse: first, the accuracy of the lowest level is quite high (above 85% for all datasets), and there is not a clear trend of increasing MCQA accuracy for increasing simulated levels. This is also shown by the scores of the monotonicity evaluation, which are in all cases worse than GPT-3.5 (0.74 for *ARC*, 0.85 for *RACE*, and 0.68 for *CUP&A*), again supporting the need for performing prompt engineering on each model. Also, this aligns with the findings from Aher et al. (2023), suggesting that advanced models such as GPT-4 might suffer of the *curse of hyper-accuracy*.

4.5 Additional analyses

The output obtained with the reference prompt contains, in addition to the index of the answer choice selected by the simulated learner, an indication of the question difficulty level (as directly assigned by the LLM) and an explanation of the answer.[11] Even though these were added during prompt engineering because they help in reaching the desired behaviour, we analyse them to understand whether they provide further insights.

4.5.1 Analysis of the difficulty level

The difficulty level assigned by LLM is on the scale $[1;5] \cap \mathbb{N}$ (the same as the simulated learners), which is different from the target difficulty values available in the three datasets. Thus, to compare them, we perform a linear scaling from $[1;5] \cap \mathbb{N}$ to the difficulty range used in each dataset ($[1;3] \cap \mathbb{N}$ in *RACE*, $[3;7] \cap \mathbb{N}$ in *ARC*, $[30;110]$ in *CUP&A*).

We find that the difficulty level provided by the LLMs cannot be directly used as an indication of question difficulty with the current prompt, and this holds true for the three datasets and the three versions of GPT we are experimenting on. The majority of questions are given difficulty values in $\{2,3,4\}$, with levels 1 and 5 almost never being assigned. Also, the LLMs are not consistent in this difficulty classification task and different difficulty levels are assigned to the same questions, possibly due to the request of simulating students of different skill levels. In terms of average error, the observed MAPE (Mean Absolute Percentage

[11] "*the list of steps that a student of level {X} would follow to select the answer, including the misconceptions that might cause them to make mistakes.*"

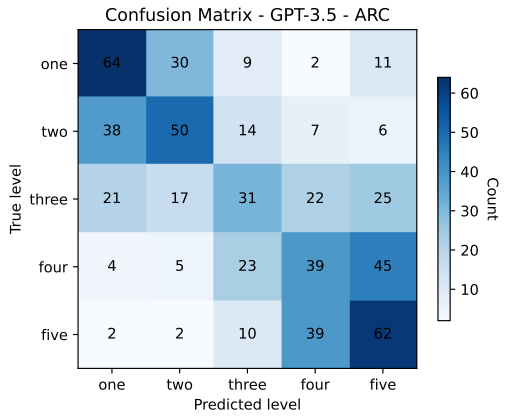


Figure 5: Confusion matrix obtained with the model for predicting the simulated level from the explanation (GPT-3.5, *ARC*).

Relevant n-grams	Simulated level *one*	*two*	*three*	*four*	*five*
	GPT-3.5				
would likely choose	186	216	66	21	5
may think that	64	30	21	7	5
however is not	18	3	6	1	1
may not understand	16	4	1	0	0
might not consider	15	4	2	0	0
would choose option	21	17	38	77	104
would understand that	8	17	44	60	49
would know that	5	22	74	142	136
misconc. that could	2	5	3	3	13
it is important	9	21	8	18	19
	GPT-4				
not fully understand	146	26	0	0	0
might not understand	50	3	0	0	0
at would likely	61	42	26	21	5
but would likely	58	26	1	0	0
might not know	47	20	1	0	0
therefore correct is	1	4	22	74	96
would not be	0	9	18	15	39
would know that	7	82	97	130	114
would recognize that	7	58	77	94	87
other option are	4	11	14	23	32

Table 2: Number of relevant n-grams in the explanations provided by the LLMs for different simulated levels, *ARC* (*misconc.* is abbreviation for *misconception*).

Error) ranges from the 12.81 for GPT-3.5-1106 on *CUP&A* to the 32.70 of GPT-4 on *RACE*. A more detailed analysis is shown in Appendix C.

4.5.2 Analysis of the explanations

We study the explanations provided by the LLMs both quantitatively and qualitatively.

For the **quantitative** analysis, we build a classifier to estimate the simulated level from the explanation provided by the LLM. A high accuracy in this level prediction task would suggest that the explanations are significantly different between the different simulated levels. To improve the interpretability of the predictor, we use TF-IDF (Manning et al., 2008) weights as features and a Logistic Regression model for the classification; the parameters of both are selected with cross-validation (parameters are listed in Appendix D.1).[12] Considering the explanations provided by GPT-3.5, the accuracy ranges from 0.41 for *RACE* to 0.55 for *CUP&A*, while considering GPT-4 it ranges from 0.52 to 0.58 (the random baseline is 0.20), suggesting that indeed there is a significant difference between the explanations for different simulated levels, and this is greater for GPT-4 than for GPT-3.5. Also, the confusions matrices show that prediction error is generally small: Figure 5 shows the confusion matrix for GPT-3.5 and *ARC*, but the trend is similar across the two models and three datasets (the other matrices are in Appendix D).

We then collect for each simulated level the most relevant n-grams, according to the prediction models, and analyse how their frequency changes across simulated levels. Table 2 focuses on the *ARC* dataset,[13] and shows the five most relevant n-grams for simulated levels *one* and *five* (for GPT-3.5 and GPT-4). Indeed the explanations often (but not always) follow a trend of using wording related to *knowledge* and *understanding* for higher levels, and the opposite for lower simulated levels. Also, i) the trend is arguably more visible for GPT-4 and ii) most of the relevant features are not shared between the two models, suggesting that the explanations are significantly different between them.

The results of this quantitative analysis show that the explanations are different for different simulated levels and they often follow meaningful patterns in term of the knowledge that learners of increasing levels are expected to have. Also, GPT-4 seems to provide better explanations than GPT-3.5, although this does not lead to a better behaviour in the simulation task, as shown in Section 4.4.[14]

We also **qualitatively** analysed some of the explanations, and indeed observed some differences

[12] Prior to this analysis, we remove any explicit reference to the simulated level from the explanation.

[13] Results for the other datasets are shown in Appendix D.

[14] This is in line with previous research suggesting that the rationales created with e.g., *chain-of-thought* are not necessarily used in the downstream task (Pfau et al., 2024).

Sim. level	*Answer explanation* from GPT-3.5
one	*A student of level one would choose the answer 'muscle cells' because they might think that muscles are responsible for feeling heat and pressure. However, the correct answer is 'nerve cells' because they are the cells that transmit signals to the brain to interpret sensations like heat and pressure.*
three	*A student of level three would select the answer 'nerve cells' because they have a basic understanding of the human body and know that nerve cells are responsible for transmitting signals related to heat and pressure. [...]*
five	*A student of level five would choose the answer 'nerve cells' because nerve cells are responsible for detecting and transmitting sensations such as heat and pressure to the brain. [...]*

Table 3: An example of how the *answer explanation* from GPT-3.5 changes depending on the simulated levels (we show only the most relevant parts of the explanations due to the lack of space). The question is: *"What helps skin feel heat and pressure?"*. The answer options: *['muscle cells', 'blood cells', 'nerve cells', 'bone cells']*.

between different simulated levels. An example is shown in Table 3, where it is visible that level *one* simulated with GPT-3.5 makes a reasonable mistake, while levels *three* and *five* get the correct answer. The full list of explanations is available in the supplementary material.

4.5.3 Analysis of different educational scales

We study the effect of using different educational scales instead of the *one* to *five* used in the reference prompt `RP`, by prompting GPT-3.5 with a modified version of `RP` to consider i) *exam marks* (*A, B, C, D, F*) and ii) a non-standardised scale (*beginner, intermediate, advanced*). In both cases, the results are very good for all datasets, leading to monotonic accuracy curves with almost linear increases (especially for *ARC*) and high scores according to the metric M (between 0.97 and 0.99), supporting the finding that LLMs might be capable of simulating different levels. The accuracy curves and prompts are shown in Appendix E.

5 Related Work

Previous research discussed the possibility of using LLMs instead of (or in addition to) human participants in surveys (Dillion et al., 2023; Argyle et al., 2023; Demszky et al., 2023), and studied whether LLMs can be prompted to show human-like behaviours in a series of task (Aher et al., 2023). However, it is not agreed whether this is actually a good practice. Indeed, some researchers argue that LLMs cannot (and should not) replace human research participants (Harding et al., 2023; Crockett and Messeri, 2023). We mostly agree with the latter, but believe that exam simulations are a different application scenario, as knowledge-based exam questions are built to assess students knowledge in an objective (as much as possible) manner. Still, possible biases of this approach will have to be studied before an application in the real world. An approach like the one proposed by Beck et al. (2023) (i.e., using LLMs as a preliminary step before the human annotations) might be adopted in education, for instance pretesting with human learners only a fraction of the original items.

Previous research also discussed profusely the potential of LLMs in education (Jeon and Lee, 2023; Kasneci et al., 2023; Caines et al., 2023). Closer to our work, previous research experimented on Knowledge Tracing with LMs (Liu et al., 2022), but without using them for simulating students. Also related to the current work is the previous research of question difficulty estimation with NLP (AlKhuzaey et al., 2023; Benedetto et al., 2023; Rogoz and Ionescu, 2024), especially when performed in an unsupervised manner (Loginova et al., 2021). Indeed, the students simulation we propose in this paper could be used as an alternative to previous approaches for difficulty estimation.

6 Conclusions and future work

In this paper, we have shown that it is possible to prompt GPT-3.5 to simulate students of different levels, and the *reference prompt* we have engineered proved capable of generalising across datasets. However, even though the prompt seems to generalise well to unseen data, it does not seem to generalise to different LLMs, thus stressing the need for prompt engineering for each model. Although we found some strong indications that it might be possible to simulate students of different levels with LLMs, there are questions still to be addressed. For a better simulation, one could try to use retrieval augmented generation (RAG) (Lewis et al., 2020) on topic specific documents to better define the level of the role-played student. For a better virtual pretesting, it will be needed to have a larger set of simulated students. Also, it might be helpful to simulate whole exams, instead of one

question at a time as we did here. Future work could also iterate on the *reference prompt*, possibly using automatic prompt optimization (Pryzant et al., 2023), and experiment with open models, which is particularly relevant since specific versions of closed LLMs can become deprecated.

7 Limitations

This work uses LLMs to simulate the responses of students to exam questions and, therefore, any decision taken upon these simulations is at risk of being biased, due to the intrinsic biases in LLMs. This risk is mitigated by the fact that exam questions are built to assess domain knowledge, but it is still present. Focusing on the aspects that are specific to the educational domain, it might happen that LLMs reproduce response patterns (and errors) only of a fraction of the population of students, similarly to how using LLMs for surveys oversamples WEIRD[15] participants (Apicella et al., 2020). If this is the case, virtual pretesting done with LLMs would not account for all the other students who make different errors. An example in language learning is the fact that students from different L1s (i.e., first language), tend to make different mistakes. If LLMs reproduce the errors of specific L1s only, this might disadvantage learners with specific backgrounds. This is a common challenge in exam item writing, and even human experts struggle with it. Possible ways to address this are i) to perform pretesting with the desired population of learners and analyse whether their responses are aligned with the ones from the models, and ii) look for biases with the *Marked Personas* approach proposed by Cheng et al. (2023).

An important point that we have raised in this paper is that the results do not seem to generalise across LLMs, as prompts which were very effective on *gpt-3.5-turbo-0613* did not work as well on *gpt-3.5-turbo-1106* and, especially, GPT-4 (*gpt-4-1106-preview*). This is a significant concern from a practitioner's perspective, since any process based on a similar approach might become unusable as soon as there is a new version of the LLM and the older one is deprecated, and suggests that moving towards open LLMs could be a better alternative.

It is worth mentioning that one of the limitations of this approach is the instability of the prompts, and the fact that minor changes to the input prompt might lead to major differences in behaviour. This is a common issue with LLMs, and could be partially mitigated by performing automatic prompt optimization as mentioned in the conclusions.

Lastly, the training dataset of GPT* models is not precisely known, and one might think that this could affect the results shown in this work. Indeed, *ARC* and *RACE* provide some information about question difficulty, and this might be leveraged in some way by the model to adapt its responses to question difficulty. We believe that it is not the case, since the *CUP&A* dataset was released very recently – it is more recent than the training data used in all the models considered in this work – and the findings are consistent across datasets.

[15]Western, Educated, Industrialized, Rich, Democratic.

Acknowledgements

This research was partially funded by Cambridge University Press & Assessment. We thank Dr. Andrea Giussani for the discussions at a very early stage of the project, as well as Dr. Roberto Turrin from QA Ltd., Hannah Bouteba and Andrew Mullooly from Cambridge University Press & Assessment, and the team at the ALTA Institute for the support. We also thank the anonymous reviewers for providing valuable feedback.

References

Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. 2023. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*, pages 337–371. PMLR.

Samah AlKhuzaey, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2023. Text-based question difficulty prediction: A systematic review of automatic approaches. *International Journal of Artificial Intelligence in Education*, pages 1–53.

Coren Apicella, Ara Norenzayan, and Joseph Henrich. 2020. Beyond WEIRD: A review of the last decade and a look ahead to the global laboratory of the future.

Lisa P Argyle, Ethan C Busby, Nancy Fulda, Joshua R Gubler, Christopher Rytting, and David Wingate. 2023. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351.

Tilman Beck, Hendrik Schuff, Anne Lauscher, and Iryna Gurevych. 2023. How (not) to use sociodemographic information for subjective NLP tasks. *arXiv preprint arXiv:2309.07034*.

Luca Benedetto. 2023. A quantitative study of NLP approaches to question difficulty estimation. In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, pages 428–434. Springer Nature Switzerland.

Luca Benedetto, Paolo Cremonesi, Andrew Caines, Paula Buttery, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. 2023. A survey on recent approaches to question difficulty estimation from text. *ACM Computing Surveys*, 55(9):1–37.

Andrew Caines, Luca Benedetto, Shiva Taslimipoor, Christopher Davis, Yuan Gao, Øistein Andersen, Zheng Yuan, Mark Elliott, Russell Moore, Christopher Bryant, et al. 2023. On the application of large language models for language teaching and assessment technology.

Myra Cheng, Esin Durmus, and Dan Jurafsky. 2023. Marked personas: Using natural language prompts to measure stereotypes in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1504–1532, Toronto, Canada. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Molly Crockett and Lisa Messeri. 2023. Should large language models replace human participants?

Dorottya Demszky, Diyi Yang, David S Yeager, Christopher J Bryan, Margarett Clapper, Susannah Chandhok, Johannes C Eichstaedt, Cameron Hecht, Jeremy Jamieson, Meghann Johnson, et al. 2023. Using large language models in psychology. *Nature Reviews Psychology*, pages 1–14.

Danica Dillion, Niket Tandon, Yuling Gu, and Kurt Gray. 2023. Can AI language models replace human participants? *Trends in Cognitive Sciences*.

Gemma Team and Google DeepMind. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Ronald K Hambleton, Hariharan Swaminathan, and H Jane Rogers. 1991. *Fundamentals of item response theory*, volume 2. Sage.

Jacqueline Harding, William D'Alessandro, NG Laskowski, and Robert Long. 2023. AI language models cannot replace human research participants. *AI & SOCIETY*, pages 1–3.

Jaeho Jeon and Seongyong Lee. 2023. Large language models in education: A focus on the complementary relationship between human teachers and chatgpt. *Education and Information Technologies*, pages 1–20.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yichan Liang, Jianheng Li, and Jian Yin. 2019. A new multi-choice reading comprehension dataset for curriculum learning. In *Asian Conference on Machine Learning*, pages 742–757. PMLR.

Naiming Liu, Zichao Wang, Richard Baraniuk, and Andrew Lan. 2022. Open-ended knowledge tracing for computer science education. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3849–3862.

Ekaterina Loginova, Luca Benedetto, Dries Benoit, and Paolo Cremonesi. 2021. Towards the application of calibrated transformers to the unsupervised estimation of question difficulty from text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 846–855.

Eric Loken and Kelly L Rulison. 2010. Estimation of a four-parameter item response theory model. *British Journal of Mathematical and Statistical Psychology*, 63(3):509–525.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge university press.

Meta. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date.

Andrew Mullooly, Øistein Andersen, Luca Benedetto, Paula Buttery, Andrew Caines, Mark JF Gales, Yasin Karatay, Kate Knill, Adian Liusie, Vatsal Raina, et al. 2023. The Cambridge multiple-choice questions reading dataset.

OpenAI. 2023. GPT-4 technical report. *ArXiv*, abs/2303.08774.

Jacob Pfau, William Merrill, and Samuel R Bowman. 2024. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.

Georg Rasch. 1961. On general laws and the meaning of measurement in psychology. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 4, pages 321–333. Univ of California Press.

Ana-Cristina Rogoz and Radu Tudor Ionescu. 2024. UnibucLLM: Harnessing LLMs for automated prediction of item difficulty and response time for multiple-choice questions. *arXiv preprint arXiv:2404.13343*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Victoria Yaneva, Peter Baldwin, Janet Mee, et al. 2019. Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–20.

A List of prompts: *Analysis of the reference prompt*

Table 4 shows the text of the four prompts that are compared to the reference prompt (`RP`) in Section 4.1 and whose behaviour is shown in Figure 1. The four prompt, similarly to `RP`, ask the LLMs to simulate students of five different skill levels (from *one* to *five*) and to produce an output in JSON format. The JSON fields that the LLMs are asked to produce are different between the prompts.

B IRT simulation

In Section 4.3 we presented the results obtained with an IRT simulation, to show an *upper threshold* for the correlation between the difficulties obtained from pretesting and the target values (i.e., the ones available in the *CUP&A* dataset) when having a very small population of five students.

IRT (Item Response Theory) (Hambleton et al., 1991) is a mathematical framework used in educational settings to estimate the latent traits of students and questions (e.g., skills and difficulties) involved in an exam. The simplest model, named "Rasch model" (Rasch, 1961), associates a skill level to each student and a difficulty level to each question; more complex models take into consideration additional latent traits (Loken and Rulison, 2010), such as the probability of correct answer by guessing.

IRT provides a function (named *item response function*) to compute the probability that a given student i correctly answers a given question j:

$$P_{correct} = c_j + \frac{1 - c_j - s}{1 + e^{-a_j(\theta_i - b_j)}} \quad (1)$$

where i) θ is the *skill* level associated to the student, ii) b the *difficulty* of the question, iii) a the *discrimination* of the question, iv) c a *guess* factor (to account for the fact that students might get the correct answer in a MCQ by randomly guessing), and v) s a *slip* factor to account for skilled students that might make mistakes due to temporary distraction or fatigue.

IRT is commonly used for pretesting exam questions (i.e., to estimate the latent traits of new items before using them to assess students). However, it can also be used, as we do in this paper, to simulate how *mock* students of known skill levels would answer questions of known difficulty. Specifically, we simulate a population of five students with skill levels uniformly distributed in the range $[30; 110]$, which is the "known" range of difficulty in *CUP&A*, answering the question in the exam. We use the following parameters for the IRT simulation: i) the difficulty values available in the *CUP&A* dataset, ii) discrimination $a = 1$, iii) guess $c = 0.25$, and iv) slip $s = 0.05$. Given these simulation parameters, we proceed as follows for all student-question pairs ij.

- We estimate the probability P_{ij} that student i will correctly answer question j according to the item response function.
- We generate a random number r uniformly distributed in $[0; 1]$.[16]
- If $P_{ij} \geq r$ we mark the question as correctly answered, otherwise we mark it as wrongly answered.
- Measure the fraction of wrong answers for each question (i.e., estimate its difficulty).

[16] Using *random.uniform* from *numpy*.

Table 4: List of prompts showed in Figure 1 in Section 4.1, where they are compared with the reference prompt. For all prompts, the student levels we consider are *['one', 'two', 'three', 'four', 'five']*.

ID	Prompt
P1	SYSTEM: You will be shown multiple choice questions from a science exam. The questions in the exam have difficulty levels on a scale from one (very easy) to five (very difficult). Similarly, the students can be identified with skill levels from one (low level student) to five (very skilled student). The level of students is defined such that a student of a certain level can answer most of the questions of lower levels, and almost none of the question of higher levels. You must assign a difficulty level to the given multiple choice question, and select the answer choice that a student of level {X} would pick. Provide only a JSON file with the following structure: {"level": "difficulty level of the question", "index": "integer index of the answer chosen by a student of level {X}", "text": "text of the answer chosen by the student"} USER: Question: "{question}" Options: "{answer options}"
P2	SYSTEM: You will be shown multiple choice questions from a science exam. The questions in the exam have difficulty levels on a scale from one (very easy) to five (very difficult). You must assign a difficulty level to the given multiple choice question, and select the answer choice that a student of level {X} would pick. Provide only a JSON file with the following structure: {"level": "difficulty level of the question", "index": "integer index of the answer chosen by a student of level {X}", "text": "text of the chosen answer"} USER: Question: "{question}" Options: "{answer options}"
P3	SYSTEM: You will be shown multiple choice questions from a science exam. The questions in the exam have difficulty levels on a scale from level one (very easy) to level five (very difficult). Similarly, each student can be given a skill level: level one represents the least skilled students, who answer most questions wrongly, and level five represents the most skilled students, who can correctly answer even the most difficult items. You must assign a difficulty level to the given multiple choice question, and select the answer choice that a student of level {X} would pick. Provide only a JSON file with the following structure: {"level": "difficulty level of the question", "index": "integer index of the answer chosen by a student of level {X}", "text": "text of the chosen answer"} USER: Question: "{question}" Options: "{answer options}"
P4	SYSTEM: You will be shown multiple choice questions from a science exam. The questions in the exam have difficulty levels on a scale from one (very easy) to five (very difficult). You must assign a difficulty level to the given multiple choice question, motivating your choice, and select the answer choice that a student of level {X} would pick. Provide only a JSON file with the following structure: {"level": "difficulty level of the question", "motivation": "reason why you assigned that difficulty level", "index": "integer index of the answer chosen by a student of level {X}", "text": "text of the chosen answer"} USER: Question: "{question}" Options: "{answer options}"

- Compute the correlation between the this difficulty and the target value available in the dataset.

C Analysis of the difficulty levels assigned by the LLMs

This section complements 4.5.1 by providing a more detailed analysis of the difficulty level directly provided by the LLM in the *"question level"* field of the output JSON.

As we have mentioned previously, both GPT-3.5 and GPT-4 are not consistent in this difficulty classification task, assigning different difficulty values to the same question, even though we are using *temperature=0*; this might be side-effect of the simulated student level mentioned in the prompt. Also, the difficulty levels output by the LLMs do not cover the whole range of levels specified in the prompt (*"...difficulty levels from one (very easy) to five (very difficult)..."*). Table 5 shows (separately for for GPT-3.5, GPT-4, and each dataset) the frequency with which each level is assigned to one of the exam questions. The table proves the inconsistency of the LLMs in performing this task, with a distribution that varies greatly when simulating different skill levels. It is worth noting that in some cases the LLMs produce outputs which are not a difficulty value in the required format; when this happens, we ignore the output for this analysis.

The fact that the difficulty values directly provided by the LLMs are not usable as a measurement of the difficulty of exam items is also shown in Figure 6 and Figure 7, which display the evaluation metrics – MAPE (Mean Absolute Percentage Error) and R2 score – obtained with the two LLMs on the three datsets.[17] Both figures show that the results are not satisfactory, with large MAPE values and R2 scores mostly negative or close to 0, and there is not a clear difference between the two models, nor between the datasets, nor between the different simulated levels (shown with different colours in the bar plot).

D Additional analysis of the explanations

This Section complements Section 4.5.2 by providing a more detailed analysis of the explanations provided by the LLMs.

[17] As mentioned in Section 4.5.1, before computing these metrics we perform a linear scaling from $[1;5] \cap \mathbb{N}$ to the difficulty range used in each dataset ($[1;3] \cap \mathbb{N}$ in *RACE*, $[3;7] \cap \mathbb{N}$ in *ARC*, $[30;110]$ in *CUP&A*).

GPT-3.5						
	Sim.	*"question level"*				
Dataset	Level	1	2	3	4	5
ARC	*one*	20	226	87	12	0
	two	17	187	104	14	0
	three	2	255	18	41	0
	four	3	152	182	12	0
	five	4	137	178	24	0
RACE	*one*	3	86	59	2	0
	two	1	69	77	2	0
	three	0	107	12	29	0
	four	0	46	102	0	0
	five	0	35	101	11	0
CUP&A	*one*	0	107	93	0	0
	two	0	73	127	0	0
	three	0	111	69	20	0
	four	0	18	182	0	0
	five	0	10	178	12	0
GPT-4						
	Sim.	*"question level"*				
Dataset	Level	1	2	3	4	5
ARC	*one*	40	261	46	1	0
	two	18	302	27	0	0
	three	16	304	29	0	0
	four	18	290	41	0	0
	five	53	272	23	0	0
RACE	*one*	26	93	29	2	0
	two	4	124	21	0	0
	three	2	91	54	3	0
	four	3	63	72	11	0
	five	8	82	47	11	0
CUP&A	*one*	0	94	100	5	1
	two	0	122	78	0	0
	three	0	42	153	5	0
	four	0	19	140	41	0
	five	0	39	127	34	0

Table 5: Distribution of the difficulty level that is assigned, in the *"question level"* of the output JSON, to each question by the LLMs (shown GPT-3.5 and GPT-4) when simulating different student levels. This table does not analyse the predictive capabilities of the LLMs, but highlights the instability of the *"question level"* field at varying simulated level.

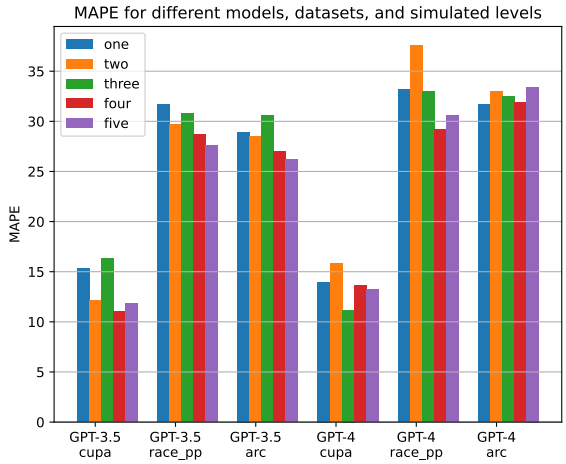


Figure 6: Mean Average Percentage Error (MAPE) evaluation of the *"difficulty level"* produced by the LLMs in the output JSON, separately for different models, datasets, and simulated student levels (the latter indicated with different colours in the bar graph). The target value is the difficulty level available in the three datasets (*grade* for *ARC*, *level* for *RACE*, and *difficulty* for *CUP&A*), and the predicted value the difficulty level output in the *"difficulty level"* field of the JSON produces by the LLMs.

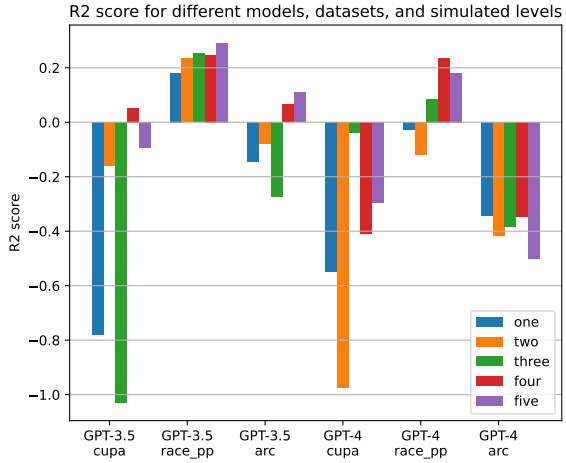


Figure 7: R2 score evaluation of the *"difficulty level"* produced by the LLMs in the output JSON, separately for different models, datasets, and simulated student levels (the latter indicated with different colours in the bar graph). The target value is the difficulty level available in the three datasets (*grade* for *ARC*, *level* for *RACE*, and *difficulty* for *CUP&A*), and the predicted value the difficulty level output in the *"difficulty level"* field of the JSON produces by the LLMs.

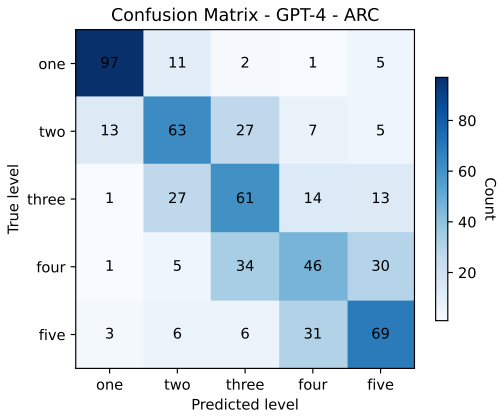


Figure 8: Confusion matrix obtained with the model for predicting the simulated level from the explanation (GPT-4, *ARC*).

Model	Dataset *ARC*	*RACE*	*CUP&A*
GPT-3.5	0.43	0.41	0.55
GPT-4	0.58	0.52	0.52

Table 6: Accuracy in the task of simulated level prediction from the explanation for the two models and the three datasets.

We start by showing the results of the evaluation of the accuracy of the prediction model trained to predict the simulated level from the explanations provided by the LLMs. The figures from Figure 8 to Figure 12 show the confusion matrices obtained for the three datasets and the two models.[18] The figures show that, in most cases, the largest values are on the diagonal (or close to it), showing that the prediction model we implemented is overall capable of correctly estimating the simulated level from the explanations. This is also supported by the prediction accuracy values we observed, which are shown in Table 6 (the random baseline is 0.20). Both results indicate that the explanations provided by both GPT-3.5 and GPT-4 are significantly different for different simulated levels.

In the main body of text we have shown which are the most relevant n-grams (according to the prediction models) for each simulated level, considering the *ARC* dataset (Table 2). The analysis showed that there seems to be a trend such that

[18]Please note that we train a separate prediction model for each dataset-LLM pair; the parameters used for GridSearchCV are shown in Section D.1.

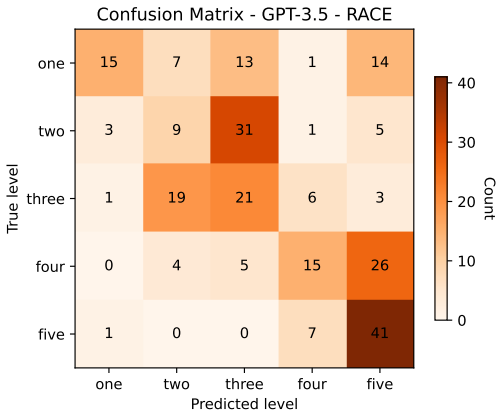


Figure 9: Confusion matrix obtained with the model for predicting the simulated level from the explanation (GPT-3.5, *RACE*).

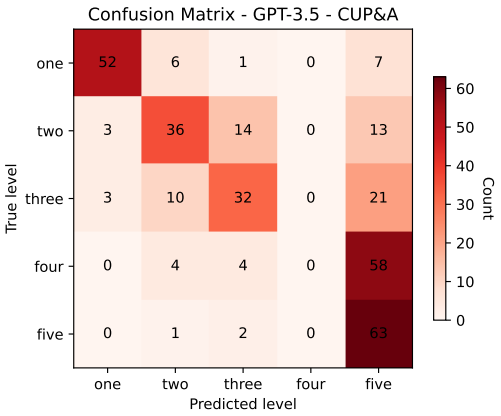


Figure 11: Confusion matrix obtained with the model for predicting the simulated level from the explanation (GPT-3.5, *CUP&A*).

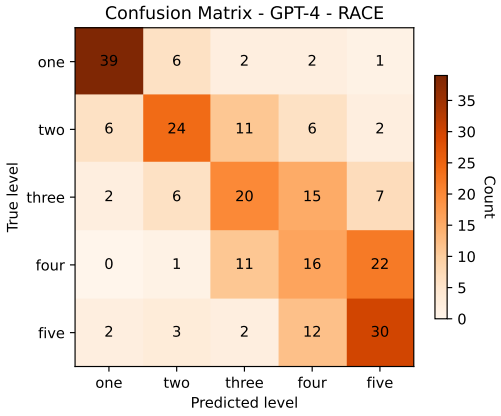


Figure 10: Confusion matrix obtained with the model for predicting the simulated level from the explanation (GPT-4, *RACE*).

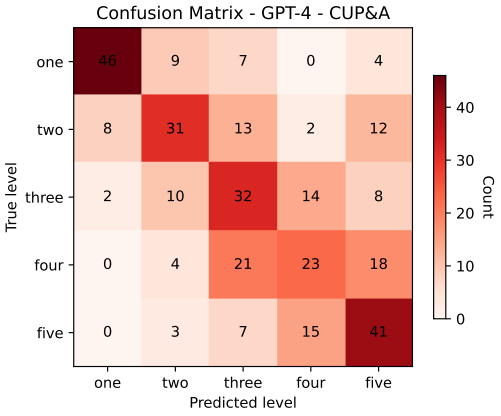


Figure 12: Confusion matrix obtained with the model for predicting the simulated level from the explanation (GPT-4, *CUP&A*).

Relevant n-grams	Simulated level *one*	*two*	*three*	*four*	*five*
	GPT-3.5				
might choose option	39	7	7	0	0
most likely choose	15	0	0	0	0
because it is	41	33	33	16	16
confused by mention	4	2	2	0	0
option that match	4	2	2	0	1
read passage carefully	0	0	0	23	40
would carefully read	0	0	0	0	13
information in passage	2	4	4	9	20
would understand that	0	0	0	10	26
would easily understand	0	0	0	16	17
	GPT-4				
might struggle with	27	1	0	0	0
might focus on	29	2	1	0	0
likely look for	44	8	0	0	0
passage directly state	11	3	2	3	5
for keywords in	26	4	0	0	0
would first identify	0	0	0	6	28
would likely not	0	1	2	9	21
is correct because	2	2	5	10	14
supported by text	11	16	21	23	21
passage and identify	1	0	8	10	8

Table 7: Number of relevant n-grams in the explanations provided by the LLMs for different simulated levels, *RACE*. We show the five most relevant (according to the prediction models) n-grams for simulated levels *one* and *five* and the two models.

Relevant n-grams	Simulated level *one*	*two*	*three*	*four*	*five*
	GPT-3.5				
might choose option	34	10	3	0	0
most likely choose	110	19	3	1	0
might think that	27	12	1	0	0
her work because	0	1	2	2	2
her what sing	1	1	1	1	1
would pick option	0	1	9	10	24
because would understand	0	0	0	0	6
he was unsure	2	0	1	1	2
he had not	1	1	1	1	1
he is concerned	1	1	2	1	1
	GPT-4				
might struggle with	80	2	0	0	0
might focus on	35	4	1	0	0
however correct is	44	14	9	4	2
struggle with abstract	58	0	0	0	0
in question and	30	6	3	1	0
would first identify	0	0	0	2	33
which is not	7	12	12	13	16
need carefully analyze	0	0	1	7	14
therefore correct is	9	17	38	48	62
other option are	16	13	19	28	33

Table 8: Number of relevant n-grams in the explanations provided by the LLMs for different simulated levels, *CUP&A*. We show the five most relevant (according to the prediction models) n-grams for simulated levels *one* and *five* and the two models.

wording related to "knowledge" is more frequent in higher simulated levels, while wording related to "uncertainty" and "mistakes" is more common in the lowest simulated level, and this is more visible with GPT-4. As a complement to that analysis, we show here the results obtained, separately for GPT-3.5 and GPT-4, for the two reading comprehension MCQs datasets: *RACE* (in Table 7) and *CUP&A* (in Table 8). In both cases, we show the five most relevant n-grams according to i) GPT-3.5 and simulated level *one*, ii) GPT-3.5 and level *five*, iii) GPT-4 and level *one*, and iv) GPT-4 and level *five*.

D.1 Parameters for GridSearchCV

The parameters used for training the predictors of simulated level from the explanations are the following:

- *'tfidf__ngram_range':[(1, 1), (1, 2), (1, 3), (2,3), (2,2), (3,3)]*,
- *'tfidf__max_df': [0.1, 0.2, 0.3, 0.4, 0.5]*,
- *'tfidf__min_df': [0.005, 0.01 ,0.015, 0.02]*,
- *'logistic__C': [0.1, 0.5, 1, 10]*,
- *'logistic__penalty': ['l1', 'l2']*.

E Analysis on different educational scales

This section complements Section 4.5.3 by showing the accuracy plots obtained with the different educational scales, the full list of scores according to the metrics to evaluation monotonicity (M), and the prompts used for these experiments. The goal of this analysis is to perform a preliminary exploration of whether it might be possible to use different educational scales from the *one* to *five* used in the reference prompt `RP`.

E.1 Exam grades (marks): A, B, C, D, F

Figure 13 shows the MCQA accuracy obtained when prompting GPT-3.5 to simulate students that got different exam grades, from A (best score) to F (worst score). The plot shows a really good behaviour across datasets, with the MCQA accuracy decreasing towards simulated students of lower skills, and it is particularly good for the *ARC* dataset. This is also shown by the scores obtained with the evaluation metric M, which are shown in Table 9. This result is particularly interesting since the LLM does not have view of the whole exam, but is given only one question at a time, without

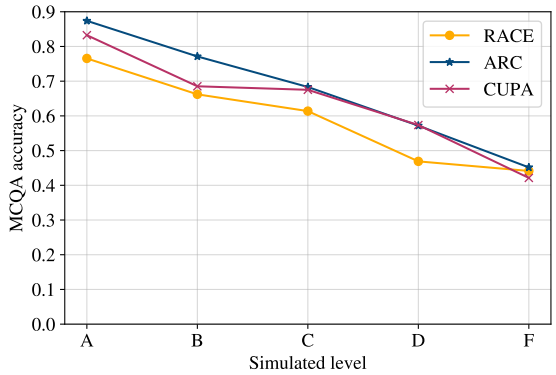


Figure 13: Evaluation of GPT-3.5 when simulating students of grade [A, B, C, D, F] on the three datasets.

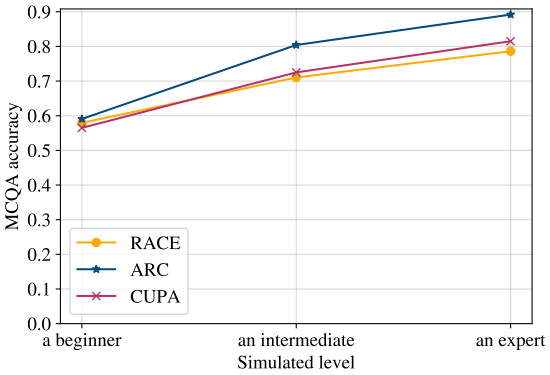


Figure 14: Evaluation of GPT-3.5 when simulating students of *beginner*, *intermediate*, and *advanced* levels on the three datasets.

	ARC	*RACE*	*CUP&A*
M score	0.99	0.98	0.97

Table 9: M scores obtained when prompting GPT-3.5 to simulate students that got different exam marks, separately on the three datasets.

any information about its responses to the other questions of the exam. The updated prompts used for this analysis are presented in Table 11, showing in bold the differences with respect to the reference prompt (RP).

E.2 Abstract scale: *beginner*, *intermediate*, *advanced*

The results obtained when prompting GPT-3.5 to simulate the student levels *beginner*, *intermediate*, and *advanced* are shown in Figure 14. For all datasets, we can observe the desired monotonic trend of increasing MCQA accuracy for increasing simulated levels, and this is also shown by the scores obtained with the evaluation metric, shown in Table 10. The updated prompts used for this analysis are presented in Table 12, showing in bold the differences with respect to the reference prompt (RP). The results obtained with these updated prompts support the finding that LLMs (specifically GPT-3.5, in our experiments) might indeed be used to simulate students of different levels, although future work is needed to precisely control the accuracy obtained with each simulated level.

	ARC	*RACE*	*CUP&A*
M score	0.97	0.98	0.98

Table 10: M scores obtained when prompting GPT-3.5 to simulate *beginner, intermediate, and advanced* students, separately on the three datasets.

Table 11: Prompts used for the experiments on exam marks analysed in Figure 13 in Section E.1. The first prompt is used on *ARC*, the second on *RACE* and *CUP&A*. In **bold** the parts that are different from the *reference prompts*.

Prompt	Student levels
SYSTEM: You will be shown a multiple choice question from a science exam, and the questions in the exam have difficulty levels on a scale from one (very easy) to five (very difficult). You must assign a difficulty level to the given multiple choice question, and select the answer choice that a **grade {X} student** would pick. Provide only a JSON file with the following structure: {"question level": "difficulty level of the question", "answer explanation": "the list of steps that a **grade {X} student** would follow to select the answer, including the misconceptions that might cause them to make mistakes", "index": "integer index of the answer chosen by a **grade {X} student**"} USER: Question: "{question}" Options: "{answer options}"	[A, B, C, D, F]
SYSTEM: You will be shown a multiple choice question from an English reading comprehension exam, and the questions in the exam have difficulty levels on a scale from one (very easy) to five (very difficult). You must assign a difficulty level to the given multiple choice question, and select the answer choice that a **grade {X} student** would pick. Provide only a JSON file with the following structure: {"question level": "difficulty level of the question", "answer explanation": "the list of steps that a **grade {X} student** would follow to select the answer, including the misconceptions that might cause them to make mistakes", "index": "integer index of the answer chosen by a **grade {X} student**"} USER: Reading passage: "{context}" Question: "{question}" Options: "{answer options}"	[A, B, C, D, F]

Table 12: Prompts used for the experiments on the additional qualitative scale (*[beginner, intermediate, advanced]*) analysed in Figure 14 in Section E.2. The first prompt is used on *ARC*, the second on *RACE* and *CUP&A*. In **bold** the parts that are different from the *reference prompts*.

Prompt	Student levels
SYSTEM: You will be shown a multiple choice question from a science exam, and the questions in the exam have difficulty levels on a scale from one (very easy) to five (very difficult). You must assign a difficulty level to the given multiple choice question, and select the answer choice that **{X} student** would pick. Provide only a JSON file with the following structure: {"question level": "difficulty level of the question", "answer explanation": "the list of steps that **{X} student** would follow to select the answer, including the misconceptions that might cause them to make mistakes", "index": "integer index of the answer chosen by **{X} student**"} USER: Question: "{question}" Options: "{answer options}"	[a beginner, an intermediate, an expert]
SYSTEM: You will be shown a multiple choice question from an English reading comprehension exam, and the questions in the exam have difficulty levels on a scale from one (very easy) to five (very difficult). You must assign a difficulty level to the given multiple choice question, and select the answer choice that **{X} student** would pick. Provide only a JSON file with the following structure: {"question level": "difficulty level of the question", "answer explanation": "the list of steps that **{X} student** would follow to select the answer, including the misconceptions that might cause them to make mistakes", "index": "integer index of the answer chosen by **{X} student**"} USER: Reading passage: "{context}" Question: "{question}" Options: "{answer options}"	[a beginner, an intermediate, an expert]