BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY
http://fit.vut.cz

Using AI to Grade Student Tests in SUI

Bc. Martin Hemza, Bc. Petr Kaška, Bc. Jakub Magda

Abstract
The aim of this project was to explore the potential of large language models (LLMs) for automatically grading short, open-ended student answers. We focused on responses scored on a scale from 0 to 4 points. Using a digitized dataset of real student answers from the SUI course, we prepared the data for model training. We experimented with two main approaches: fine-tuning an open-source models and prompt engineering using the GPT-4 API. Our results show that both methods have their strengths—while the fine-tuned model produced consistent scores similar to those of human raters, GPT-4 with a carefully designed prompt achieved comparable accuracy without any training. This project demonstrates that automatic grading with LLMs is feasible even with limited resources, and that prompt design can play a key role in performance. The findings could be valuable for schools, teachers, and developers of digital assessment tools.

Keywords: LLM, GPT, fine-tuning, automatic evaluation, student tests

Faculty of Information Technology, Brno University of Technology

1. Introduction

Grading short written answers from students can be time-consuming and sometimes inconsistent. Teachers often have to read many different responses that express similar ideas in very different ways. With the rise of large language models (LLMs) like ChatGPT, there is growing interest in using AI to help with this process. These models can understand and generate human-like text, which makes them promising tools for evaluating student answers.

Artificial intelligence (AI) is being increasingly used in education, especially for tasks where automation could reduce teachers' workload and provide students with faster feedback. In particular, automatic grading of open-ended questions is an area where AI could make a real impact. However, relying on AI systems for such tasks also raises important questions about fairness, transparency, and accuracy.

In this project, we explored whether a language model can automatically assign scores to student responses on a scale from 0 to 4 points. We worked with real answers from previous exams, which were converted into digital form. Some parts of the data had to be cleaned or fixed by hand to make sure everything matched correctly. Then, we tried two main approaches: training our own model and using existing ones through prompt engineering.

Our goal was not just to get high accuracy, but also to understand what works best when using AI for grading. Unfortunately, the wider deployment of such technologies is hampered by strict EU regulations (e.g. the proposed EU AI Act[1]), which aim to ensure responsible and safe use of AI. Despite these challenges, our work demonstrates the potential of language models as educational tools.

This paper presents what we learned, what worked well, and where there's still room for improvement.

2. Task

The primary objective of this paper is to develop a system capable of automatically evaluating test answers using a grading scale ranging from 0 to 4 points per response.

To establish a reliable baseline, we fine-tuned a multilingual BERT model using a standard classification setup. The input consisted of a concatenation of the question and answer texts, and the model was trained to predict one of 17 discrete labels correspond-

[1] https://artificialintelligenceact.eu/ai-act-explorer/

ing to scores from 0.0 to 4.0 in 0.25 increments. The predicted class labels were then mapped back to numeric scores, allowing us to evaluate regression accuracy using Mean Absolute Error (MAE).

Despite the simplicity of this approach, it provided a surprisingly strong baseline. The final model achieved a **MAE of 1.64** on the test set.

3. Dataset

The dataset used in our project was obtained from students in previous years, who had manually converted handwritten exams into digital format using *Optical Character Recognition (OCR)*. However, the dataset contained minor inconsistencies in matching answers to their corresponding questions. To address this, we manually reviewed and corrected the pairings.

Additionally, the question texts themselves were scanned and processed via *OCR*, which introduced grammatical and typographical errors, even when referring to the same question. To ensure consistency, we normalized and standardized the wording of all questions. This step reflects a realistic deployment scenario, where the set of questions would be predefined and manually entered, eliminating the need for scanning.

Some questions lacked corresponding student answers. These were excluded from the dataset, as in practical use, such cases would be unambiguous and not require automatic scoring. We also removed questions that were not text-based in nature—such as those requiring drawings or mathematical formulas—as our model focuses exclusively on textual input.

3.1 Selected Dataset

Due to the computational demands of the fine-tuning process, we did not use the entire dataset. Instead, we selected a representative subset consisting of responses to ten distinct questions.

Figure 1 presents a histogram showing the number of responses per selected question. Figure 2 displays the distribution of scores for each question using box plots, offering insight into how frequently each score occurs per question.

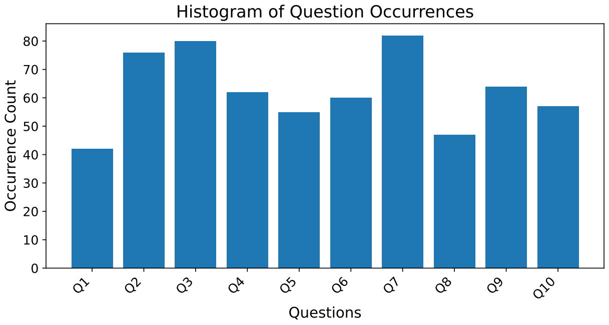

Figure 1. Number of Responses per Question

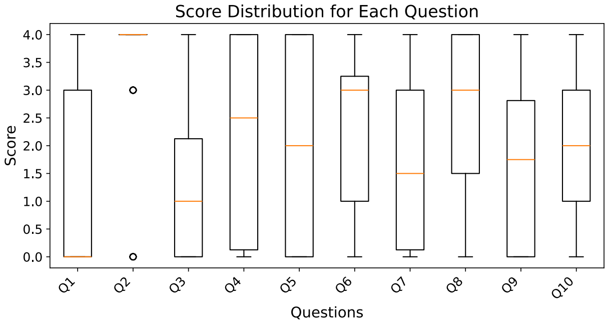

Figure 2. Score Distribution per Question

4. Methods

This section describes the two methods we used to aid in the evaluation.

4.1 Fine tuning models

To adapt a language model for the task of automatic grading, we employed parameter-efficient fine-tuning using the *LoRA (Low-Rank Adaptation)* technique. The input was formatted as a prompt-completion pair, where the prompt consisted of the question and a student's answer, and the expected completion was the assigned score.

Given our computational constraints, we limited our experiments to models with fewer than 10 billion parameters. To ensure compatibility with Czech-language data, we selected models that demonstrated understanding of Czech according to the *BenCzechmark*[3]. We tested several models available through Hugging Face that performed reasonably well in Czech across various tasks.

For memory efficiency, models were loaded using 4-bit quantization. *LoRA* was applied to a subset of layers to reduce the number of trainable parameters while maintaining effectiveness. Fine-tuning was conducted using the *SFTTrainer* from the *trl* library, configured for ten epochs and tracked via *Weights & Biases (W&B)*. Evaluation was carried out by comparing the model's predicted scores with human-assigned scores using regression metrics such as **Mean Absolute Error (MAE)** and **Mean Squared Error (MSE)**.

4.2 GPT API

In this experimental phase, we deliberately avoided interfering with the model architecture and instead relied entirely on prompt-engineering with the pre-trained GPT-o4-mini[6], GPT-o3-mini[5] and the modified dataset (see 3.1).

In the first method, in zero-shot mode, we presented the model with only the assignment text and the student answer, without providing any reference examples, and let it independently assign a score in the range 0-4 (it had the option to score half points).

In the second method, we switched to a one-shot approach, where we added a template of 9 sample answers to the prompt with corresponding teacher ratings (each 0-4 score category was represented by one example), and then added this template to all other student answers for which we again obscured the ratings, thus demonstrating to the model the quality levels of the answers corresponding to each score. This allowed us to compare the pure zero-shot performance of the model with the results enriched by the benchmark examples, and to quantify how prompt-engineering without any adjustment to the weighting parameters affects the accuracy and consistency of the scores.

In the third method, implemented in collaboration with Mr. Hradiš, for each question, we first divided the expected answer into several evaluation sub-criteria - for example, the first point for stating a basic fact, the second point for expanding it with additional related information, and the third point for a deeper interpretation or example. These explicit rubrics were put together to cover all relevant levels of response quality, and then submitted with the student responses to the model, which automatically assigned points based on the defined criteria. We also tried to explicitly tell the model to rate Mr. Hradish's column moderately.

In the fourth experiment, we constructed a large-scale reference prompt that included all available questions as well as student responses from previous years for each question-answer from dataset. Then, for ten randomly selected question in the prompt, we hid its teacher rating and asked the model to estimate the hidden sample's score on its own based on the ground-truth residual and we did this for each score. In this way, we gave GPT-o4-mini the ability to leverage the rich context of historical examples in generating consistent and accurate scoring.

Unfortunately, after discarding many non-fulltext questions and filtering out poorly parsed questions using OCR techniques, we were left with only a small fraction of questions for experimentation.

5. Results

In this section we will discuss the results of the individual experiments.

5.1 GPT API

Prompting experiments were performed mainly on one model, o4-mini[6], because it is the latest mini model from OpenAI. However, we also tried one experiment on another model, o3-mini[5], which has fewer parameters than the previous model. We wanted to test whether the fact that the model is "dumber" means that it will tend to assess students more leniently or more severely. We evaluate the results for each experiment in two graphs, the first tells how much the model underestimates/overestimates for each question. And the second graph indicates how many points it missed for each question.

The results of the o4-mini model in the First experiment are shown in the Figure 7. Average errors are around 1.5-2 points and without significant outliers. For a purely prompting model that did not have the correct question wording or reference solutions, the model performance is quite good.

In Experiment 2, shown in Figure 8 where the model was additionally given a sample question with both the student's answer and their score, the bias moved closer to zero for all questions, with the largest value dropping from about -2 points to about -1.0 points (for the question). The median absolute error decreased for most questions. The error variance is also slightly smaller, but there are still outliers around 4 points. So, even though the example of a scored answer does help (less systematic underestimation and lower mean error), the model still underestimates scores and occasionally makes large errors.

In Experiment 3 in Figure 9, we presented the o4-mini model with the question, the student's answer, and the assessment rubric we received from Mr. Hradiš in the first stage. In the next phase, we added kindness (i.e., we incorporated a request to be extremely accommodating into the prompt), the results are in Figure 10. The last phase we tested the o3-mini model with kindness, Figure 11. By default, the o4-mini model systematically underestimated students across all five questions, with median absolute errors lying around 1-2 points and isolated fluctuations ranging up to 4 points as shown on Figure 9. Then, on Figure 10 when we added the considerate prompt to the o4-mini prompt , the bias paradoxically deepened and median absolute error rose to with more widespread variance.

We applied the same “kindness” to o3-mini and the bias shifted to an interval of approximately -0.5 to -1.7 points and the mean errors rose to 2-3 points, albeit with slightly narrower bins than for o3-mini. Results are in Figure 11. Thus, the addition of the kind prompts did not lead to generous (over-) estimation, but instead deepened the underestimation and increased the errors. The o3-mini maintains only slightly more stable results than the o4-mini in this setting.

In Experiment 4, show in Figure 12, (calibration based on all other students’ scores), the behavior of the o4-mini model improved significantly compared to previous runs. While the bias for the first four questions is still slightly below zero, the underestimation is significantly smaller compared to Experiment 2 and Experiment 1. Conversely, in the second half of the questions (Q5-Q9), the model now overestimates slightly, so that overall the bias fluctuates only within a narrow band of about ±0.8 points.

5.2 Fine tuning models

We fine-tuned three Czech-capable language models: **Mistral-7B-Instruct-v0.3**[4], **CSTinyLlama-1.2B**[1], and **Czech-GPT-2-XL-133k**[2]. Each model was evaluated using identical settings and datasets, allowing us to compare their performance fairly.

The results for **Mistral-7B** are shown in Figure 4. While the model performed extremely well on the training set after ten epochs—achieving near-perfect loss—it exhibited signs of overfitting. Both MAE and MSE increased on the testing set over time, indicating poor generalization.

In contrast, **CSTinyLlama-1.2B** (see Figure 5), which has significantly fewer parameters than Mistral, achieved the lowest performance of the three. However, it still outperformed a simple baseline, demonstrating that even small models can capture useful patterns in scoring tasks.

The best-performing model was **Czech-GPT-2-XL-133k**, as shown in Figure 6. Despite having fewer parameters than Mistral, it achieved the lowest **MAE (0.994)** and **MSE (2.046)**, indicating a better fit to the task without overfitting. This suggests that moderate-sized, Czech-trained models can be highly effective for automated short-answer grading.

In Figure 3, we visualize the per-question accuracy of the Czech GPT model during its best-performing epoch. When compared with the score distributions in the dataset (see Figure 2), it becomes apparent that questions with lower variance in human-assigned scores were predicted more accurately by the model.

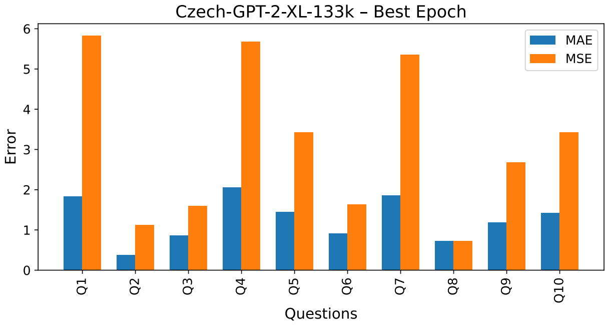

Figure 3. Number of Responses per Question

6. Conclusion

Prompting (from basic polling to sample rubric to leave-one-out calibration) reduces bias and average error by 4-mini, but biases of ±1 point and occasional errors >3 points persist. Full teacher replacement therefore requires human verification, yet the semi-automated workflow saves time and standardizes scores.

Fine-tuning of smaller Czech models (CSTinyLlama-1.2B, Czech-GPT-2-XL-133k) outperformed Mistral-7B-Instruct and achieved better accuracy without overfitting, confirming the benefit of domain-fitted models over pure prompting.

References

[1] BUT FIT. *CSTinyLlama-1.2B*. 2024. Accessed: 8.5.2025. Available at: https://huggingface.co/BUT-FIT/CSTinyLlama-1.2B.

[2] BUT FIT. *Czech-GPT-2-XL-133k*. 2024. Accessed: 8.5.2025. Available at: https://huggingface.co/BUT-FIT/Czech-GPT-2-XL-133k.

[3] FAJCIK, M., DOCEKAL, M., DOLEZAL, J., ONDREJ, K., BENES, K. et al. BenCzechMark: A Czech-centric Multitask and Multimetric Benchmark for Language Models with Duel Scoring Mechanism. 2024. Accessed: 8.5.2025. Available at: https://huggingface.co/spaces/CZLC/BenCzechMark.

[4] MISTRAL AI. *Mistral-7B-Instruct-v0.3*. 2024. Accessed: 8.5.2025. Available at: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3.

[5] OPENAI. *OpenAI o3-mini*. 2025. Accessed: 8.5.2025. Available at: https://api.openai.com/.

[6] OPENAI. *OpenAI o4-mini*. 2025. Accessed: 8.5.2025. Available at: https://api.openai.com/.

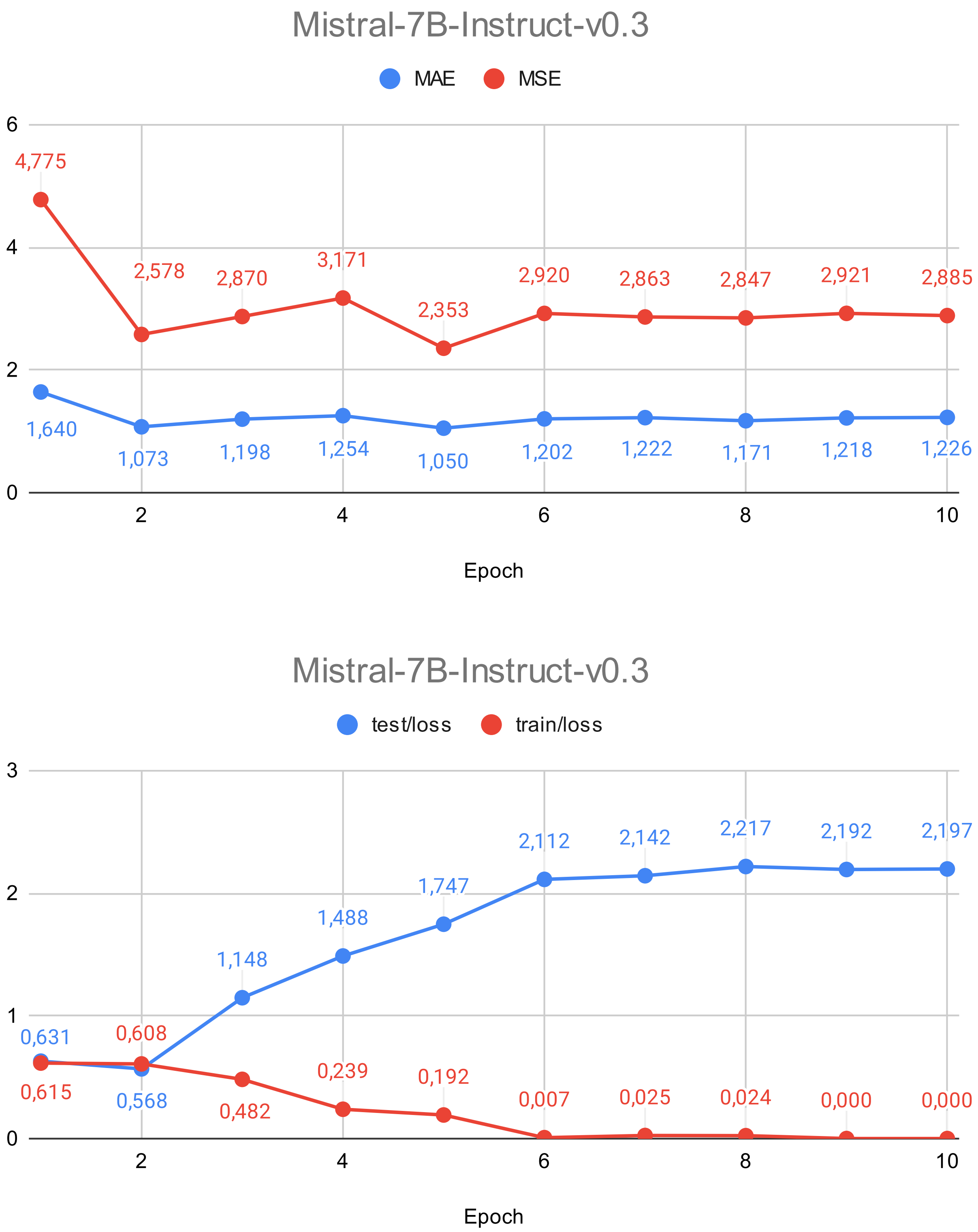

Figure 4. Fine-Tuning **Mistral-7B-Instruct-v0.3**[4]

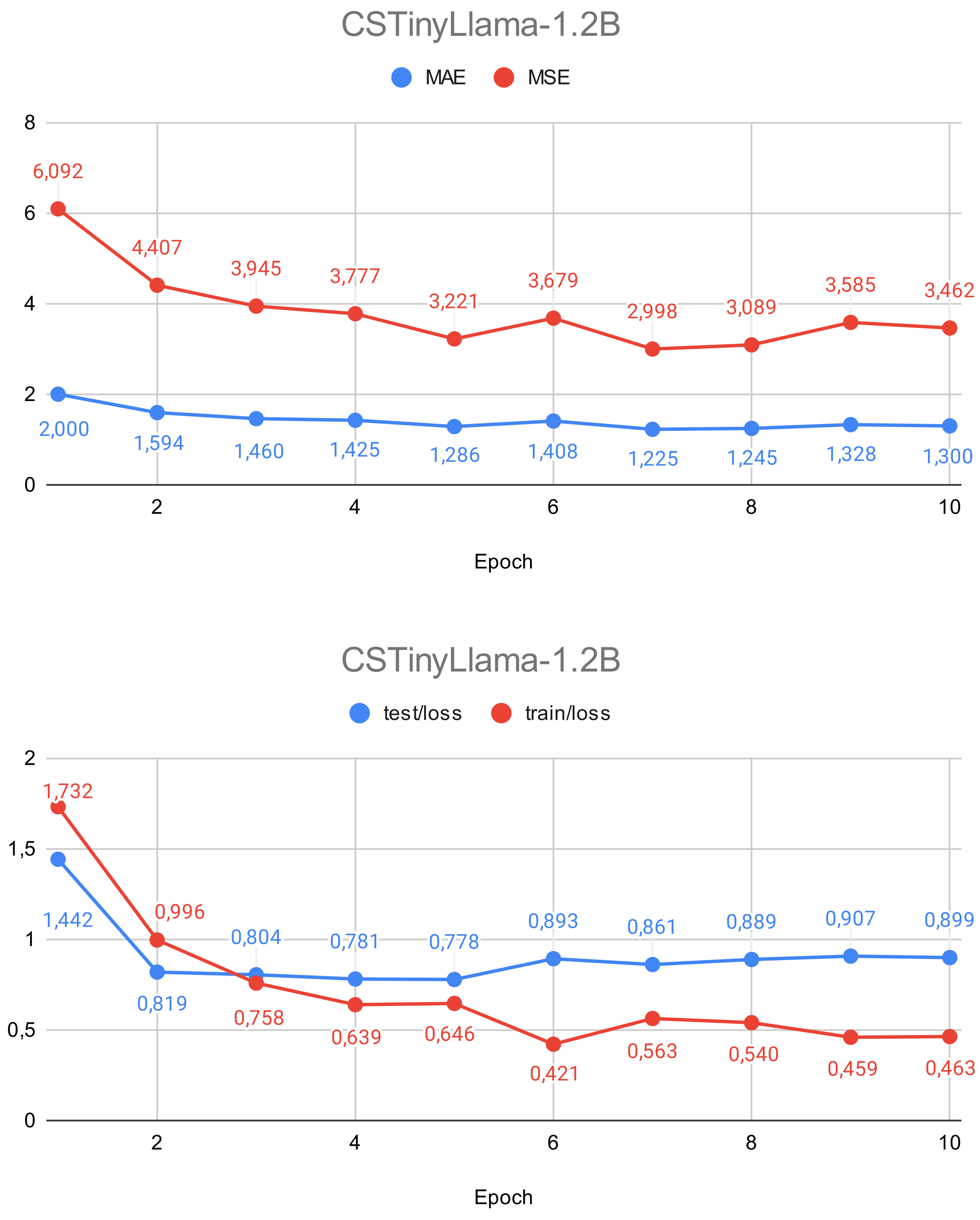

Figure 5. Fine-Tuning **CSTinyLlama-1.2B**[1]

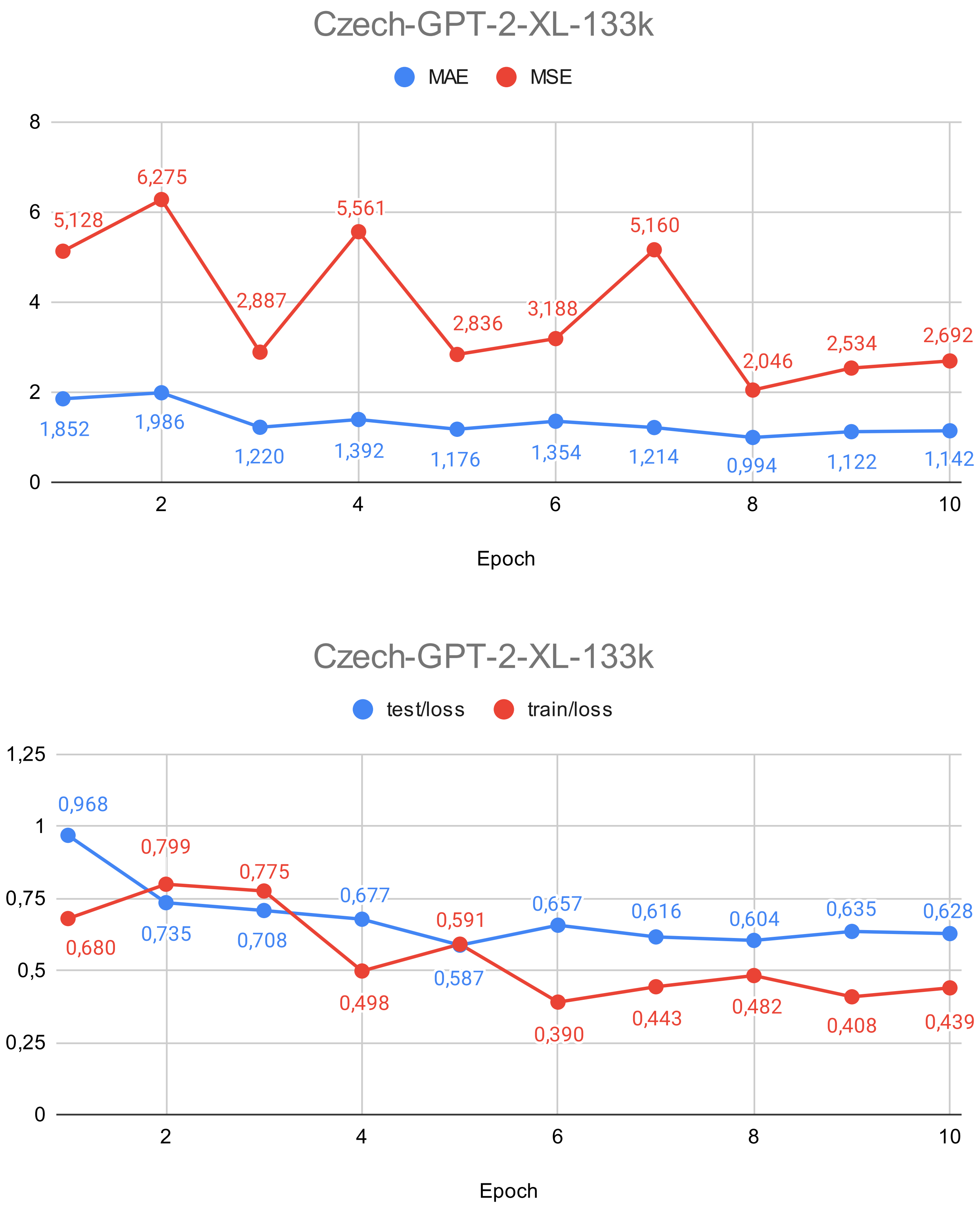

Figure 6. Fine-Tuning **Czech-GPT-2-XL-133k**[2]

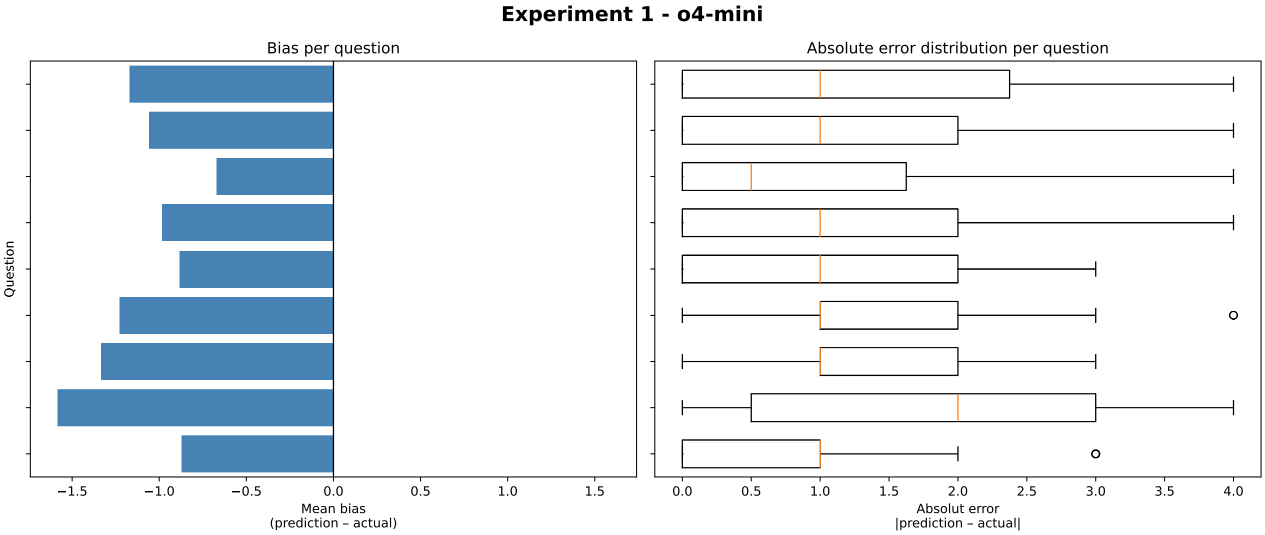

Figure 7. Prompting GPT model o4-mini [6]

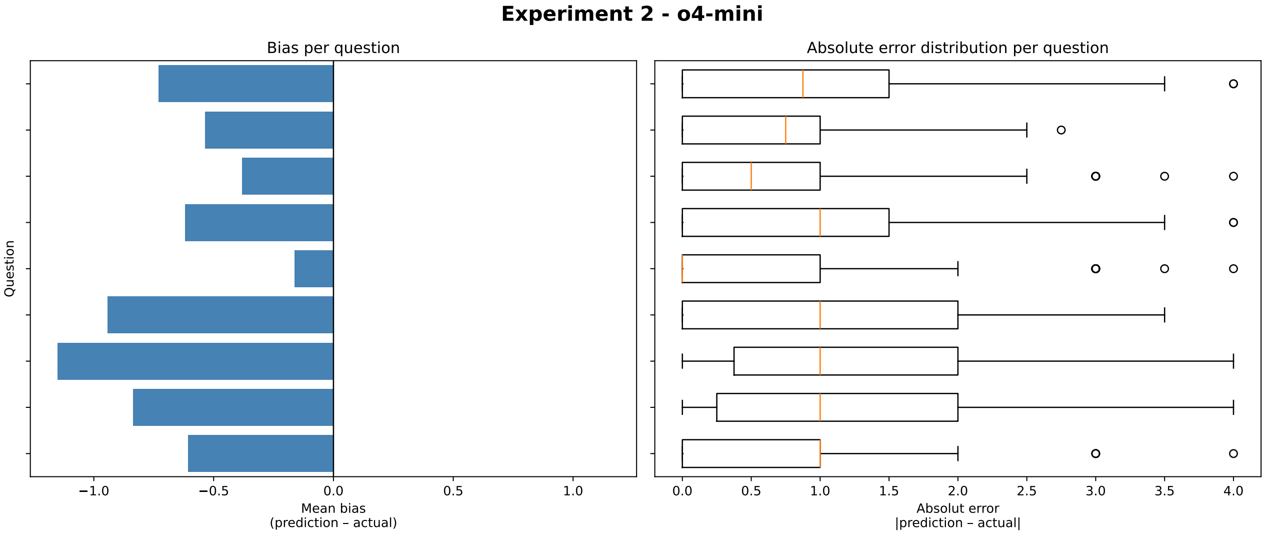

Figure 8. Prompting GPT model o4-mini [6]

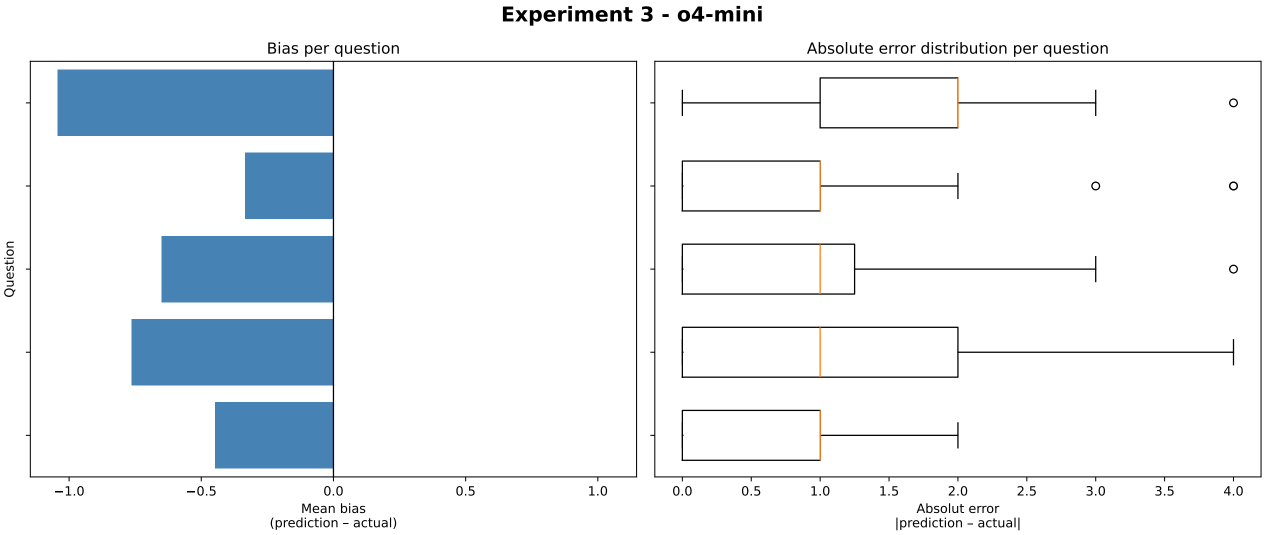

Figure 9. Prompting GPT model o4-mini[5]

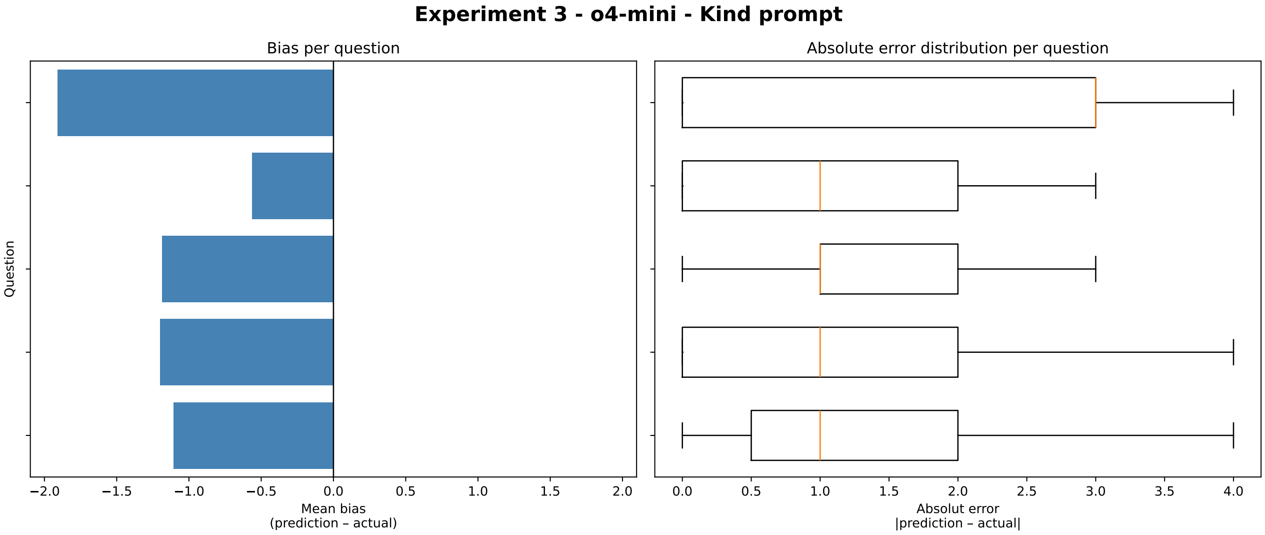

Figure 10. Prompting GPT model o4-mini[6] - Kind prompt

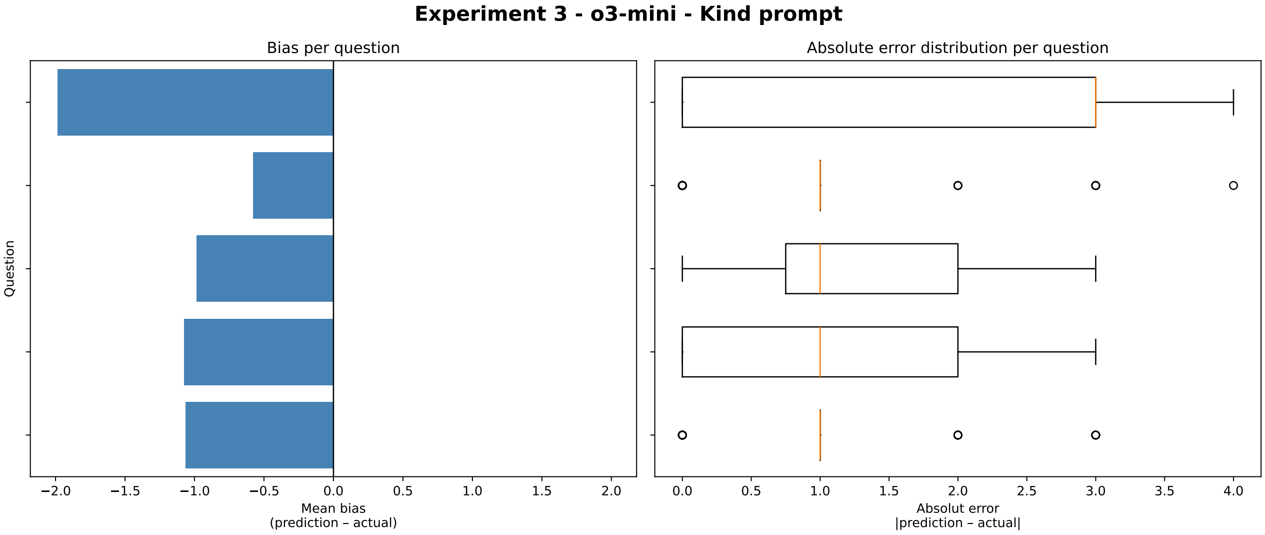

Figure 11. Prompting GPT model o3-mini[5] - Kind prompt

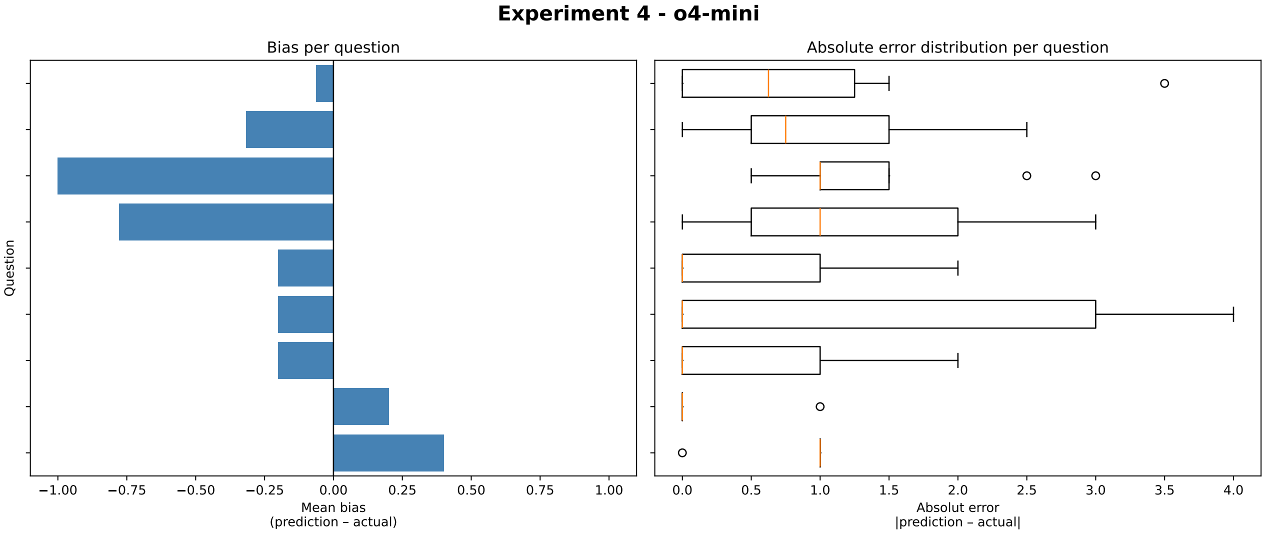

Figure 12. Prompting GPT model o4-mini[5]