Alpaca-Tutor: A LLM-Based Chatbot for Assisting Students in STEM Courses

Adriana Orellana | 376792 | adriana.orellanatorrico@epfl.ch
Angel Zenteno | 376890 | angel.zenteno@epfl.ch
Renqing Cuomao | 377052 | renqing.cuomao@epfl.ch
Alpaca-AAR

Abstract

This paper presents Alpaca Tutor, a specialized AI assistant designed to support college students in STEM fields. Current large language models struggle to address the needs of college education, particularly in areas that require complex mathematical and reasoning concepts. To overcome this limitation, we created a new dataset using large language models with strong reasoning skills, such as GPT4-o, Llama 3 70B, and Gemini Flash. We generated synthetic data by providing these models with chain-of-thought examples and asking them to generate step-by-step reasoning and add the phrase "Final Answer:" at the end of their responses. Our approach involves three main steps: enhancing the EPFL preference pairs dataset, fine-tuning our base model, and applying Direct Preference Optimization (DPO) to align its responses with human preferences. We used the enhanced dataset to fine-tune Llama 3 8B and then applied DPO to align its responses with human preferences. Our evaluation results show that the format "Final Answer:" is capable of returning the correct option for multiple-choice answers. Alpaca Tutor provides accurate and step-by-step solutions to complex problems, making it a valuable resource for college students in STEM fields.

1 Introduction

Natural language processing has allowed the development of chatbots and instruct models that can perform various tasks, such as summarization, translation, and text generation. These models have become highly useful in fulfilling general user necessities. However, most of them struggle with mathematical or reasoning problems, especially those that involve complex concepts. This limitation is particularly evident in areas like college education, where the complexity of the questions requires a deeper understanding of mathematical and reasoning concepts.

Despite the improvements in recent large language models, they still struggle to address the needs of college education, especially when it comes to smaller models. One of the primary reasons for this limitation is the lack of data at the university level. As a result, when students seek help with complex concepts, coding, or math problems using these instruct models, they often encounter erroneous answers. It is because one known problem of large language models is hallucination, which can lead to misleading information (Xu et al., 2024).

Creating new datasets is a helpful step to solve this issue. However, most of these datasets focus on high school education, which may not be sufficient to meet the needs of college students (Amini et al., 2019; Wu et al., 2023). Moreover, a critical requisite for new datasets is to provide the reasoning behind their solutions. This information is essential to guide students towards a better understanding of the problems they need help with.

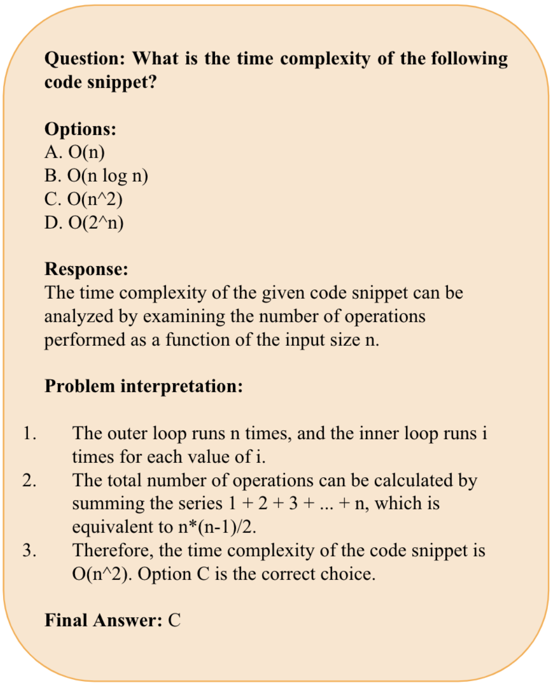
In our approach, we were initially provided with preference pairs annotated by students of the MNLP class using GPT-3.5-turbo. However, since the answers were not provided, we could not guarantee the quality of this dataset. Therefore, we decided to create a new dataset. Due to the lack of annotators and the time-consuming nature of this task, we leveraged large language models with strong reasoning skills, such as GPT4-o, which is currently the best LLM for reasoning. We labeled the GPT4-o answers as our ground truth and generated additional answers with Llama 3 70B and Gemini Flash. Inspired by a previous work (Kim et al., 2023), since these models are larger than GPT-3.5, we assumed that their reasoning skills and answers are of better quality.

To ensure consistency, we asked the models to provide step-by-step reasoning and add the phrase "Final Answer:" at the end of their responses. This approach not only facilitates the extraction of the option selected for multiple-choice questions but also enables our model to answer multiple-choice questions after applying Direct Preference Optimization (Rafailov et al., 2023) without requiring additional fine-tuning. After completing our synthetic dataset, we created a new preference pairs dataset by selecting the preferred pairs from the larger models and rejecting those from the student-annotated pairs. We will discuss special cases in the following sections. Next, we implemented the necessary steps for Direct Preference Optimization (DPO) to align our model with human preferences.

After evaluating our specialized Llama 3 8B model, we showed that the format "Final Answer:" is already capable of returning the correct option for multiple-choice answers. Then, we experimented with quantization, which demonstrated that reducing the model's size not only decreases memory footprint but also has a minimal impact on performance. This paper is structured as follows: Section 2 provides an overview of existing research in the field, highlighting similar AI tutors and their limitations. Section 3 explains the details of our approach, describing how we generate synthetic data to build our AI tutor. Section 4 presents the experimental design, including the baselines, datasets, and evaluation methods used to quantify our results. Section 5 offers a qualitative analysis of our model's performance, highlighting the key findings. Section 6 explores the ethical implications of our work, considering the potential consequences of deploying AI tutors in educational settings. Finally, Section 7 concludes the paper, summarizing our contributions.

2 Related Work

Solving complex reasoning problems, such as mathematical or STEM questions, requires a systematic approach to arrive at accurate solutions. However, many llms struggle with this challenge, often generating answers without providing the underlying reasoning. To address this limitation, researchers have explored fine-tuning pre-trained models using specialized datasets designed to tackle reasoning problems. For instance, the Minerva model (Lewkowycz et al., 2022) was trained on a vast dataset of scientific papers and mathematical expressions, employing a chain of thought approach to enhance its performance. This approach has shown significant improvements in model performance. However, building a tutor assistant capable of handling complex math and STEM problems is constrained by the lack of data covering undergraduate or graduate-level topics.



Question: What is the time complexity of the following code snippet?

Options:
A. O(n)
B. O(n log n)
C. O(n^2)
D. O(2^n)

Response:
The time complexity of the given code snippet can be analyzed by examining the number of operations performed as a function of the input size n.

Problem interpretation:

1. The outer loop runs n times, and the inner loop runs i times for each value of i.
2. The total number of operations can be calculated by summing the series 1 + 2 + 3 + ... + n, which is equivalent to n*(n-1)/2.
3. Therefore, the time complexity of the code snippet is O(n^2). Option C is the correct choice.

Final Answer: C

Figure 1: Example provided to the models to generate answers applying step-by-step reasoning, and adding "Final Answer:" at the end of their generations

Existing math datasets, such as those providing challenging competition mathematics problems (Hendrycks et al., 2021), are limited in scope. Recent studies have proposed new datasets comprising book chapters related to STEM topics (Chevalier et al., 2024), but these require costly annotation to be extended. To overcome this issue, synthetic datasets (Ding et al., 2024) have been proposed, leveraging LLMs with promising reasoning capabilities to generate new datasets. As a result, this approach reduces data labeling time.

Building on these insights, we propose an approach to generate a new preference pairs dataset using questions from the EPFL dataset and text generated by GPT4-o, Llama 3 70B, and Gemini Flash 1.5. By applying Direct Preference Optimization (DPO) to this dataset, we aim to enhance the reasoning capabilities of Llama 3 8B, aligning its responses with human preferences and enabling it to provide more effective assistance to students, which includes step-by-step solutions.

3 Approach

In this section, we introduce Alpaca Tutor, an AI assistant that answers university STEM questions with step-by-step reasoning. Our approach involves three main steps: enhancing the EPFL preference pairs dataset, fine-tuning our base model, and applying Direct Preference Optimization (DPO) to align the model with human preferences.

Enhancing the EPFL Preference Pairs Dataset

We started with the EPFL preference pairs dataset, which consisted of labeled student answers. However, we found that the answers lacked a consistent format and contained errors. We manually checked the content of the dataset and noticed that there was no agreement in the format of the answers, and many of them seemed incorrect. Moreover, since we did not have access to the ground truths, we could not compute metrics to determine the quality of this dataset.

Inspired by the idea that larger models generate better answers, we decided to create a new dataset using bigger models like GPT4-o, Llama 3 70B, and Gemini Flash. We used the OpenAI API[1] to generate 5 answers for each question in the EPFL preference pairs dataset using GPT4-o. We did the same using Llama 3 70B using the API that Replicate[2] offers, and Gemini Flash using Vertex[3]. Joining all the answers from these models, we obtained 15 answers for each question.

We provided these models with chain-of-thought examples in the format shown in Figure 1. Our objective was to create a new EPFL dataset that is more consistent in format. Moreover, the "Final Answer:" will allow us to extract easily when we face multiple-choice answers.

However, after the generation, we noticed that GPT4-o answers were longer than the other models Llama 3 70B and Gemini Flash. To avoid the model easily figuring out which are the chosen preference pairs, we decided to paraphrase GPT4-o answers to have a similar length in all the answers. This process will be explained in more detail in Section 4.

Supervised Fine-tuning

Our second step involves performing supervised fine-tuning as it is a prerequisite before applying DPO, to ensure the data we train on is in-distribution for the DPO algorithm. For this, our datasets were the GPT4-o without paraphrasing, math QA, and STEM QA. We used parameter-efficient fine-tuning using Low-Rank Adaptation (LoRA) (Hu et al., 2021) and Unsloth[4], a library that provides LoRA implementation. Fine-tuning was done by 3 epochs, using a learning rate of 2e-4 and a LoRA rank of 16. Additionally, we applied the LoRA modules in the projection layers: query, key, value, and output, as well as gate, up, and down.

Direct Preference Optimization

After we built our new EPFL preference pairs dataset and performed supervised fine-tuning, we started training our model with DPO. The process is illustrated in Figure 3.

The idea of using DPO is that using preference data composed by a context that in this case is a question, we have a good response which is considered the chosen answer, and a bad response which is the rejected response. DPO uses a loss function that considers the likelihood of a chosen response over a rejected response and optimizes the large language model toward that objective. The policy used for this loss function is displayed in Equation 1, where x is the context, y_w is the preferred or chosen response, and y_l is the rejected response.

$$\mathrm{L}(\pi_\theta; \pi_{\text{ref}}) = -\, E_{(x,y_w,y_l)\sim D} \Big[\log \sigma \Big(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \Big)\Big] \tag{1}$$

We trained our model with DPO using the Unsloth DPO implementation, which allowed us to optimize the model performance on the preferred data. Specifically, we trained the model for 1 epoch with a learning rate of 5e-6, the value of this parameter is the one recommended by default for the library. After completing the training process, we evaluated the model's performance on the test data to assess its ability to answer both multiple-choice questions and open-ended questions. Additionally, we applied quantization to this model to reduce its memory footprint and evaluated its performance. The results of both evaluations are presented in the following section, where we provide a detailed analysis of the model's performance.

[1] https://platform.openai.com/docs/models/gpt-4o

[2] https://replicate.com/

[3] https://cloud.google.com/vertex-ai

[4] https://github.com/unslothai/unsloth

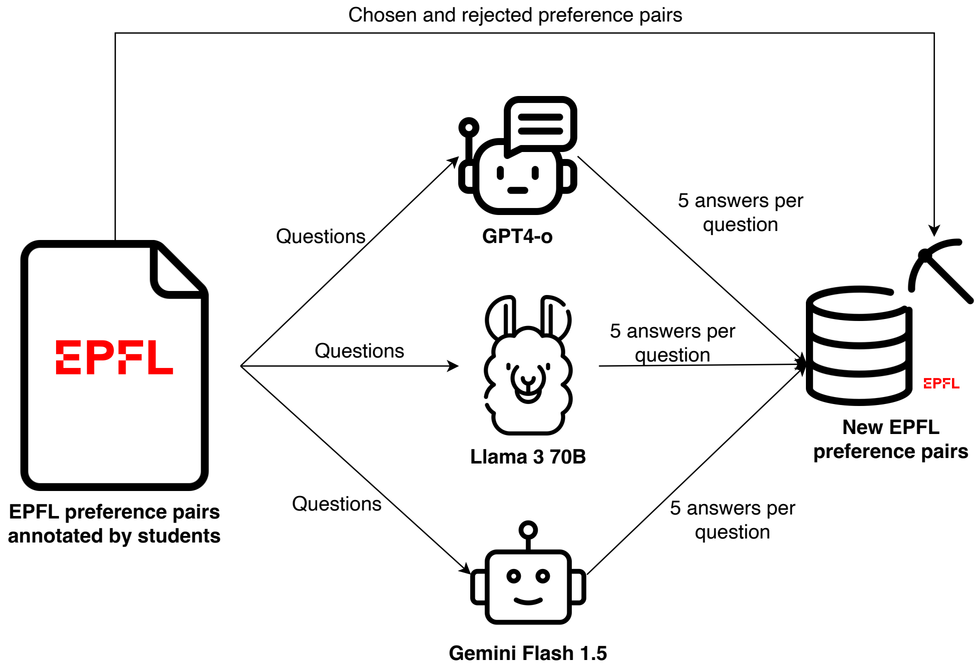


Figure 2: Process to generate new EPFL preference pairs dataset

4 Experiments

This section presents the data used to train the Alpaca Tutor, the evaluation methodology, and the baseline models used as a point of reference for performance. Finally, we present the results achieved after applying DPO and quantization, highlighting the model's strengths and weaknesses.

- **Data:** We used the following datasets to train Alpaca Tutor:
  MATH: It contains 12,500 challenging competition mathematics problems, along with the reasoning behind each problem (Hendrycks et al., 2021). We pre-processed them to create multiple-choice options and added the correct option following the phrase "Final Answer:". An illustrative example can be found in Section A.1 of the Appendix. This dataset was used for fine-tuning Llama 3 8B.
  STEM qa: Inspired by TutorEval[5], a dataset containing 834 chapters of STEM graduate courses, we used GPT4-o to generate five multiple-choice questions per chapter. As a result, we created a new dataset that consists of questions, options, and correct answers, with an example provided in Section A.2 of the Appendix. This dataset was used for fine-tuning Llama 3 8B.
  The remaining three datasets were generated using various models to provide diverse answers for the same questions. Specifically, we used GPT-4o to generate five answers each for the original EPFL dataset, resulting in a total of 7590 samples. The original GPT-4o dataset consists of too long answers so it was used for fine-tuning Llama 3 8B and after we paraphrased it we used it for DPO training. We also used Llama 3 70B and Gemini Flash 1.5 to generate five answers each for the same EPFL dataset, resulting in two additional datasets with 7590 samples each. Each sample includes the reasoning behind the chosen answer, accompanied by the phrase "Final Answer:" and the correct option letter. Thus, these datasets were used for DPO training where the preferred data comes from these bigger models, and the rejected samples come from the rejected pairs of the old EPFL preference pairs labeled by students.

Paraphrased GPT-4o SFT dataset

The primary issue identified with our gener-

[5] https://huggingface.co/datasets/princeton-nlp/TutorEval

ated data was the high similarity between answers, making it easy for DPO (Direct Preference Optimization) to distinguish between preferred and rejected pairs. This reduced the effectiveness of the dataset for training purposes, as the lack of variability undermined the quality and diversity of the generated responses. To address this, we aimed to increase the variation and improve the quality of the data by paraphrasing the answers while preserving their original meanings.

To achieve the desired variability, we implemented a paraphrasing process using the Meta LLaMA 3 70B model through the Replicate API. For each answer, the paraphrasing model was prompted to rephrase the text, maintaining the original information but varying the wording, sentence structures, and idea sequences. This ensured that each paraphrased response conveyed the same meaning in a different form, while ensuring that the answer and reasoning is still correct. We employed a range of temperatures (0.1 to 0.8) to control the randomness of the model's output, enhancing the diversity of the generated paraphrases. The maximum length for each paraphrased answer was set to twice the average length of the original answers to ensure thorough rephrasing.

The paraphrased dataset shows substantial variation in the expression of similar ideas, mitigating the original problem of high similarity between responses and providing a more diverse set of responses, making it more suitable for training robust models. The variation in wording and structure helps prevent overfitting and improves the generalization capabilities of models trained on this data.

- **Evaluation method:**

  The evaluation methodology combined quantitative measures (Rouge, BLEU, Accuracy, F1) with semantic assessment (BERTScore) to provide a comprehensive view of our model performance. By using these metrics, we were able to evaluate the model's ability to generate correct answers and high-quality reasonings.

  To handle multiple reference explanations, each generated explanation was individually scored against all five references for Rouge, BLEU, and BERTScore. We then averaged these scores to obtain a final metric for each explanation. This approach ensured that the evaluation captured the overall lexical and semantic alignment with the range of acceptable responses. By averaging the results, we accounted for the diversity of references, resulting in a more robust evaluation of model performance.

- **Baselines:**

  We evaluated the performance of LLMs - Llama 3 8B instruct, Phi3 mini, mistral 7B on the paraphrased GPT-4o SFT dataset and compared the performance of our improved model with chosen base version which outperformed in zero-shot evaluation. Accuracy, F1, BLEU, ROUGE, and BERTScore metrics were used to evaluate model predictions and reasoning quality for multiple-choice questions and BLEU, ROUGE, BERTScore were used for open-ended questions in evaluation of base models.

  We used a systematic prompting technique to guide the model, requesting it to choose the correct answer and provide detailed reasoning in an exact format without any predefined information(Singh et al., 2024).

  The detailed evaluation results on gpt-4o generated multiple-choice questions (MCQAs) shows the base model's solid understanding of the semantics behind the questions and reasonings, as indicated by the high bertscore. However, Llama struggles with producing text that closely matches the reference in terms of the exact wording and order, and it shows moderate accuracy on multiple-choice questions. Future efforts should be focused on improving the model's ability to generate text with greater lexical similarity to the references and improving its performance over a wider range of responses.

  Mistral 7B showed moderate overlap with the reference texts, achieving Rouge1 scores of 0.40 and 0.41, and low BLEU scores, indicating limited n-gram similarity. In comparasion, Phi3 Mini showed slightly better lexical similarity with a higher Rouge1 score of 0.46 for MCQAs, but matched Mistral's semantic performance with BERTS scores around 0.85.

  LLaMA's overall higher accuracy and balanced performance across other metrics make

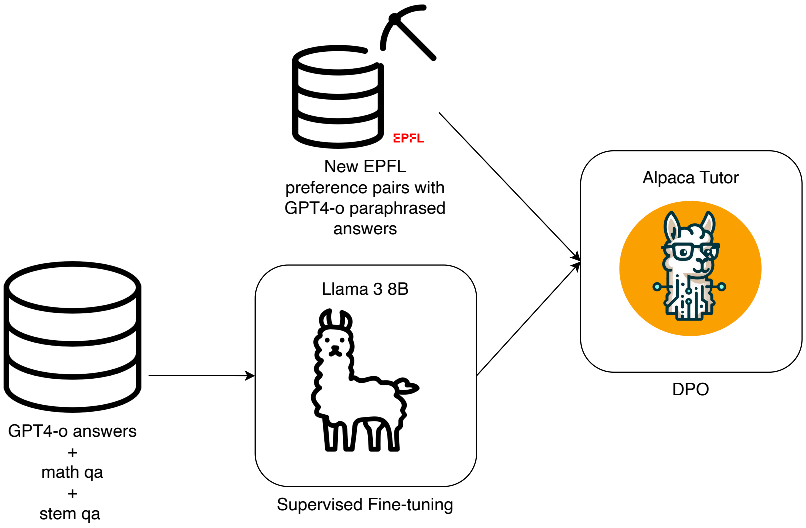


Figure 3: Process to train Alpaca Tutor

Model	Accuracy	F1	RougeL	BERTScore	BLEU
Llama3 instruct	0.48	0.47	0.27	0.86	0.04
Phi	0.33	0.37	0.27	0.86	0.06
Mistral	0.37	0.38	0.23	0.85	0.03
Alpaca tutor	0.47	0.43	0.27	0.86	0.07
Alpaca tutor quantized	0.47	0.44	0.29	0.86	0.08

Table 1: Evaluation results for GPT-4o Preference pairs with zero-shot.

it the preferred choice for fine-tuning. Its strengths in both capturing the semantic content and accurately identifying correct answers provide the foundation for further improvement in complex language understanding tasks.

- **Experimental details:**

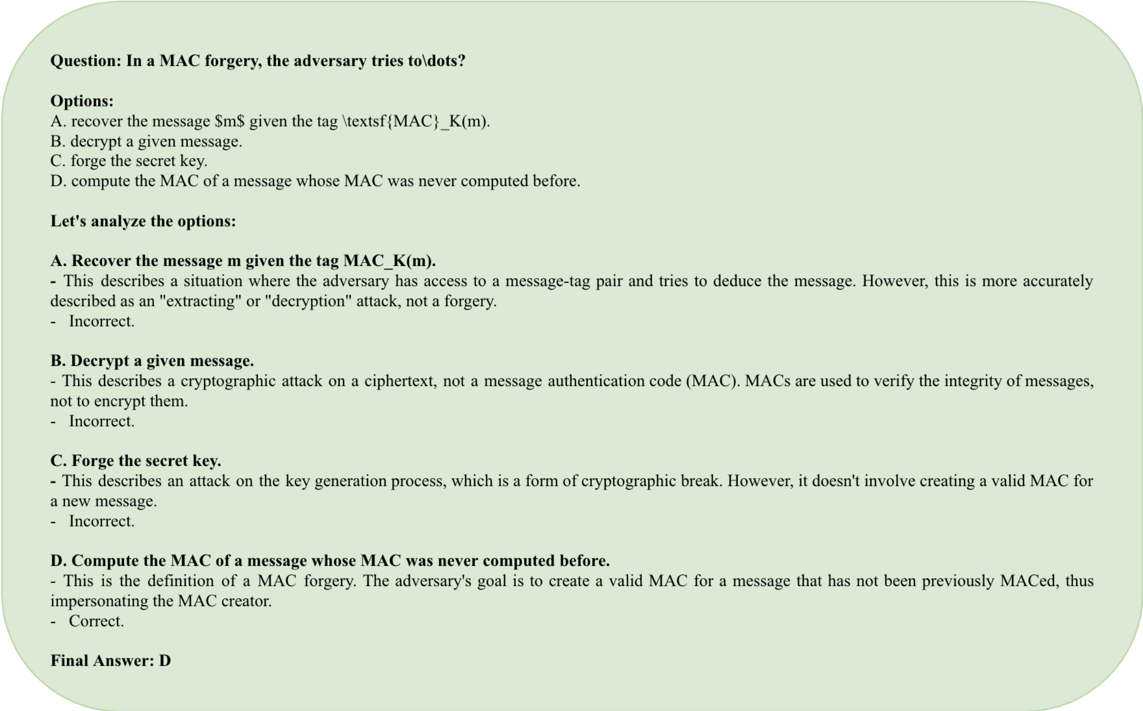
  In our experiments, we guided the model to generate accurate answers and explanations using a structured prompt and fine-tuned parameters. We set the temperature to 0.7 and top-p to 0.8, trying different learning rate to balance randomness and coherence in the generated responses. These parameters helped produce diverse yet relevant answers, ensuring high-quality, contextually appropriate explanations.

- **Results:**

  Finally, after fine-tuning and optimization, our final model, Alpaca Tutor, and its quantized version, Alpaca Tutor Quantized, showed improved performance over the base models on our EPFL preference pairs dataset. Both versions achieved high accuracy and balanced F1 scores, with Alpaca Tutor Quantized showing slight improvements in RougeL (0.29) and BLEU (0.08), indicating better lexical and semantic alignment. Table 1 shows the evaluation results of models on paraphrased gpt-4o generated EPFL Preference Pairs with zero-shot.

5 Analysis

Our analysis of the Alpaca-Tutor chatbot's performance reveals that it excels in theory-based questions, leveraging its extensive training on 15 trillion tokens of data to accurately understand and apply theoretical concepts. This is evident in its correct response to the MAC forgery question, where it demonstrated a strong grasp of cryptographic concepts, showcasing its ability to differentiate be-



Question: In a MAC forgery, the adversary tries to\dots?

Options:
A. recover the message m given the tag \textsf{MAC}_K(m).
B. decrypt a given message.
C. forge the secret key.
D. compute the MAC of a message whose MAC was never computed before.

Let's analyze the options:

A. Recover the message m given the tag MAC_K(m).
- This describes a situation where the adversary has access to a message-tag pair and tries to deduce the message. However, this is more accurately described as an "extracting" or "decryption" attack, not a forgery.
- Incorrect.

B. Decrypt a given message.
- This describes a cryptographic attack on a ciphertext, not a message authentication code (MAC). MACs are used to verify the integrity of messages, not to encrypt them.
- Incorrect.

C. Forge the secret key.
- This describes an attack on the key generation process, which is a form of cryptographic break. However, it doesn't involve creating a valid MAC for a new message.
- Incorrect.

D. Compute the MAC of a message whose MAC was never computed before.
- This is the definition of a MAC forgery. The adversary's goal is to create a valid MAC for a message that has not been previously MACed, thus impersonating the MAC creator.
- Correct.

Final Answer: D

Figure 4: Example of correct prediction of Alpaca Tutor

tween various types of attacks and identify the correct definition of a MAC forgery attack. The chatbot's performance in this question highlights its strength in understanding and applying theoretical knowledge, which is a critical aspect of academic support. However, the chatbot struggles with questions that involve arithmetic or complex mathematical operations, as seen in its failure to calculate the correct offset in the adversarial example question. This weakness is likely due to the model's lack of training on mathematical operations and its reliance on pattern recognition rather than mathematical reasoning. The chatbot's inability to perform mathematical calculations accurately, such as taking gradients and normalizing vectors, is a significant limitation that needs to be addressed. Furthermore, the chatbot may lack common sense or real-world experience, which can lead to inaccuracies or incomplete responses, particularly in scenarios where contextual understanding is crucial. Additionally, the chatbot may overfit to its training data, which can result in poor generalization to unseen questions or scenarios. Moreover, its limited domain-specific knowledge in certain areas, such as cryptography and machine learning, can lead to inaccuracies or incomplete responses. To improve the chatbot's performance, we plan to incorporate mathematical reasoning, increase domain-specific knowledge, and enhance its common sense and real-world understanding. By addressing these limitations, we can develop a more comprehensive and accurate chatbot that provides effective academic support to students.

6 Ethical considerations

The use of our AI tutor model raises several important considerations. One potential concern is that students may rely too heavily on the model's step-by-step answers, potentially damaging their ability to think critically and solve problems. Additionally, these types of assistants are not immune to errors, and students should be encouraged to carefully review and verify the accuracy of the responses. Educators should also be mindful of the potential for cheating and design assessments that promote authentic learning.

Furthermore, adapting our AI tutor model to different languages and communication methods introduces critical ethical considerations. For high-resource languages like French and German, accurate translations and culturally relevant content are necessary to ensure effective and respectful tutoring. For low-resource languages such as Urdu or Swahili, the model's effectiveness depends on developing robust transfer learning techniques and collaborating with native speakers to create valuable educational datasets.

To support signed language users, our model would need integration with sign language recognition and generation systems. This adaptation requires significant advancements in visual processing and the creation of comprehensive sign language datasets, taking into account regional variations in sign language.

7 Conclusion

In conclusion, we have presented Alpaca Tutor, an AI assistant designed to provide step-by-step reasoning and accurate solutions to university STEM questions. Our approach leverages large language models with strong reasoning skills, such as GPT4-o, Llama 3 70B, and Gemini Flash, to generate a new preference pairs dataset that addresses the limitations of existing datasets. By fine-tuning our base model, Llama 3 8B, using this dataset, and applying Direct Preference Optimization, we have aligned our model with human preferences, enabling it to provide more effective assistance to students. Our evaluation results demonstrate the model's strengths in theory-based questions but also highlight its weaknesses in arithmetic and complex mathematical operations. To further improve the model's performance, we plan to incorporate mathematical reasoning, increase domain-specific knowledge, and enhance its common sense and real-world understanding. By addressing these limitations, we can develop a more comprehensive and accurate AI tutor that provides effective academic support to students.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *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 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexis Chevalier, Jiayi Geng, Alexander Wettig, Howard Chen, Sebastian Mizera, Toni Annala, Max Jameson Aragon, Arturo Rodríguez Fanlo, Simon Frieder, Simon Machado, Akshara Prabhakar, Ellie Thieu, Jiachen T. Wang, Zirui Wang, Xindi Wu, Mengzhou Xia, Wenhan Jia, Jiatong Yu, Jun-Jie Zhu, Zhiyong Jason Ren, Sanjeev Arora, and Danqi Chen. 2024. Language models as science tutors.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Luu, and Shafiq Joty. 2024. Data augmentation using llms: Data perspectives, learning paradigms and challenges.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Yoo, and Minjoon Seo. 2023. Aligning large language models through synthetic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13677–13700, Singapore. Association for Computational Linguistics.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Simranjit Singh, Georgios Pavlakos, and Dimitrios Stamoulis. 2024. Evaluating zero-shot gpt-4v performance on 3d visual question answering benchmarks.

Haoyi Wu, Wenyang Hui, Yezeng Chen, Weiqi Wu, Kewei Tu, and Yi Zhou. 2023. Conic10K: A challenging math problem understanding and reasoning dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore. Association for Computational Linguistics.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models.

A Appendix

A.1 MATH sample

Question: *Given that* $a - b = 5$ *and* $a^2 + b^2 = 35$, *find* $a^3 - b^3$.

Options:

- A. 100
- B. 150

- C. 175
- D. 200

Output: We know that $(a-b)^2 = a^2 - 2ab + b^2$. Therefore, we plug in the given values to get $5^2 = 35 - 2ab$. Solving, we get that $ab = 5$. We also have the difference of cubes factorization $a^3 - b^3 = (a-b)(a^2 + ab + b^2)$. Plugging in the values given and solving, we get that $a^3 - b^3 = (5)(35+5) = (5)(40) = \boxed{200}$.

Final Answer: D

A.2 STEM QA sample

Question: *Why should regression lines not be used outside the scope of the original observations?*

Options:

- A. Because they can only describe linear relationships.
- B. Because extrapolation can lead to misleading conclusions.
- C. Because they are only valid for the data used to create them.
- D. Because they require random assignment of x levels.

Output: Regression lines should not be used outside the scope of the original observations because extrapolation can lead to misleading conclusions. This is because the relationship described by the regression line may not hold outside the range of the data used to create it.

Final Answer: B