

# A chatbot for education at EPFL

(team RFL: Félicie Giraud-Sauveur, Romane Clerc, and Louise Font)

**Félicie Giraud-Sauveur**

`felicie.giraud-sauveur@epfl.ch`

## Abstract

The goal of this project is to use language models to collect data and train reward models for producing a ChatGPT agent in the area of education assistance at EPFL. Building a chatbot for education is a major challenge, as language models perform well in producing consistent outputs but that can be completely factually wrong. To meet our objective, we have fine-tuned the pre-trained DeBERTa reward model and the pre-trained conversational "cedp-sam/chatbot\_fr" model with our data. The data comes from questions/answers from EPFL courses or answers generated by students in the CS-552 course using chatGPT. External datasets have also been added. In both cases the results are unsatisfactory. The reward model training did not learn from fine-tuning as the loss did not decrease, and the final chatbot did not produce good quality outputs. However, our data and our approach aim to offer a first attempt at the problem.

## 1 Introduction

Recently, Natural Language Processing (NLP) and Machine Learning techniques have played a pivotal role in enabling interactive and personalized learning environments. In this project, we aim to train an AI tutor using language models to provide education assistance at EPFL.

The primary objective of our project is to develop an AI tutor capable of answering questions from courses at École Polytechnique Fédérale de Lausanne (EPFL). We approach this task by leveraging supervised fine-tuning data, a reward model, and generative pre-trained large language models (LLMs). The ultimate goal is to create a chatbot that can assist students in their learning journey by providing accurate and helpful responses.

The challenge lies in ensuring the AI tutor's ability to generate contextually relevant and accurate answers. Although existing language models have achieved remarkable performance in various NLP

tasks, they often struggle in educational domains due to the specific nuances and technicalities associated with academic material. Thus, it becomes essential to develop methodologies that can effectively prompt the language model and improve its performance in an educational context. This is all the more important given that there are few critical points in the use of LLMs in education. LLMs are complex black box models, and this hinders the identification of errors, biases, and discriminatory elements within the model. LLMs can also generate hallucination or the generation of erroneous output because they lack understanding of text context and meaning. This poses a risk of producing incorrect information that sounds correct, especially in specialized or uncommon topics, which is crucial to consider in an educational setting. Furthermore, LLM output can contain biases and discriminatory elements, further complicating their use [1].

We divide our project into three key steps. First, we collect supervised fine-tuning data by prompting a state-of-the-art language model, ChatGPT, with a curated set of questions from the EPFL course. In the second step, we train a reward model using the collected data and external data. This reward model serves as an evaluator, enabling us to rank and compare different responses generated by the AI tutor. Finally, in the third step, we fine-tune a generative pre-trained language model using supervised learning. We evaluate the quality of the text generations produced by the model using the developed reward function.

To build our reward model, we used the pre-trained reward model DeBERTa which we fine-tuned with our data using a loss comparing two interactions of different quality. The results of the training are not particularly satisfactory, as the loss does not decrease. For the final model, we used the pre-trained conversational "cedp-sam/chatbot\_fr" model that we fine-tuned with our data. The results are not very conclusive here either, as the chatbot's

responses are not consistent or satisfying. Our work therefore still needs to be greatly improved, but it's already a start. All the code and data can be found at this link <https://github.com/CS-552/project-m3-rfl>.

The remainder of this paper is structured as follows: Section 2 provides a literature review, discussing relevant work in the field. Section 3 details the methodology and approaches we use to fine-tune the reward model and the final model. In Section 4, we present the experiment with data, experimental details, evaluation strategy and results. Finally, Section 5 concludes the paper, highlighting the limitations and potential future directions.

## 2 Background

Chatbots, also known as conversational agents, have been a significant presence in human-computer interaction for over half a century. Some well-known examples include ELIZA, ALICE, and SmarterChild. In the field of education, chatbots have served as conversational teaching agents since the early 1970s. They have proven to be valuable tools for facilitating learning by providing instant access to course content, assignments, rehearsal questions, and study resources. Furthermore, chatbots can engage with students individually or support collaborative learning activities. Chatbots have been proposed in a variety of fields such as computer science, languages, general education, engineering, and mathematics. While most chatbots are accessible via web platforms, a few are available on mobile and desktop platforms. In the educational realm, chatbots assume different roles, including teaching agents, peer agents, teachable agents, and motivational agents. Teaching agents, emulating human instructors, can deliver instructions, provide examples, pose questions, and offer immediate feedback. Conversely, peer agents foster peer-to-peer interactions by serving as learning companions [2].

Concerning chatbot design principles, a distinction can be made between personalized learning chatbots, which adapt educational content based on individual learning weaknesses, styles, and needs, and chatbots that employ experiential learning, social dialogue, collaborative learning, affective learning, learning by teaching, or scaffolding techniques [2]. However, it is important to note that this overview dates from July 2022, and significant progress has been made since then. Currently, the

most prominent chatbot in the education domain is ChatGPT.

Developed by OpenAI and launched in November 2022, ChatGPT has garnered widespread attention for its remarkable performance in generating consistent, systematic, and informative responses. It quickly became the fastest-growing user app in history, amassing 100 million active users by January 2023, a mere two months after its release. ChatGPT serves as a virtual tutor for students, aiding them in their learning endeavors. It fulfills a dual function, acting as a learning companion by answering questions, summarizing information, and facilitating collaboration, while also serving as an assessment tool by checking understanding, preparing for exams, assisting with writing, and providing feedback [3].

Despite ChatGPT's impressive capabilities, several challenges persist, particularly regarding accuracy and reliability. It can sometimes rely on biased data, possess limited up-to-date knowledge, and generate incorrect or false information. For example, ChatGPT can provide information that is well written but totally inaccurate. Moreover, it lacks the ability to accurately assess its own accuracy. The accuracy and reliability of ChatGPT has been questioned in a number of fields, such as mathematics, sports science and psychology, as well as the health professions [3].

The realm of education often involves intricate data and necessitates the utilization of factual information. Tasks such as correctly converting formulas, performing numerical applications, and conducting reasoning that not only appears coherent but is also factually accurate present challenges for ChatGPT. It struggles to deliver accurate responses in these scenarios. However, there is an opportunity to enhance its performance by generating answers that appear relevant but are also verifiably accurate. This can be achieved by assigning confidence labels to solution provided by ChatGPT to refine a reward model, enabling the identification of correct answers amidst incorrect ones. Such an approach would yield tangible benefits in this field. While our project does not currently delve into Reinforcement Learning from human feedback (RLHF), it remains the ultimate goal: to create a chatbot that consistently provides genuinely relevant, accurate, and coherent answers. Achieving this objective would signify a significant advancement in the development of chatbot technology.

### 3 Approach

#### 3.1 Reward Model training approach

Our reward model is based on the "OpenAssistant/reward-model-deberta-v3-large-v2" pre-trained reward model. This is a "Reward model (RM) trained to predict which generated answer is better judged by a human, given a question." [4]. We fine-tuned this reward model with our data using a customized version of this model and its tokenizer. The principle of the fine-tuning of this pre-trained model is presented in Fig.1 and is inspired by the *Training language models to follow instructions with human feedback* paper [5].

The input is a batch that contains three tuples. These tuples are the combination of the three interactions what corresponds to three different answers for the same question  $x$ . The choice of putting all the interactions concerning the same question in the same batch comes from the given paper. The authors explain that they found that if we simply shuffle the comparisons into one dataset, a single pass over the dataset caused the reward model to overfit. Whereas, doing the training on all  $\binom{K}{2}$  comparisons from each prompt as a single batch element, solved the problem. Each tuple is composed of two interactions, each interaction being in the following format {'entry id':..., 'sol id':..., 'label':..., 'chat': 'System:... \n \n Human:... \n \n Assistant:... \n \n Human:... \n \n Assistant:... etc.', 'source':...}. The label is a score between 0 and 6 that indicates the quality of the answer of the assistant. In the tuples, the two interactions are ordered by label.

The CustomRewardModelTokenizer extends the tokenizer of the DeBERTa model. It takes the chats of the inputs and organized it in two variables: a question and an answer. The question is one string that groups all the texts of the 'System' and the 'Human', and the answer is one string that groups all the texts of the 'Assistant.'. Then the CustomRewardModelTokenizer use the tokenizer of the DeBERTa model to encode the question and answer variables.

Then our CustomRewardModel takes as input the encoded question and answer of the bad\_interaction and the good\_interaction. In the following we will use the notation  $(x, y_l)$  for the bad\_interaction and  $(x, y_w)$  for the good\_interaction. Then it apply the DeBERTa model to get a score  $r_\theta$  for each one. The model

outputs  $\sigma(r_\theta(x, y_w) - r_\theta(x, y_l))$ .

The loss is then calculated with the following formula:

$$\text{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x, y_w, y_l) \sim D} [\log(\sigma(r_\theta(x, y_w) - r_\theta(x, y_l)))]$$

where  $K=3$  as we have 3 interactions for one question  $x$ .

Then we backpropagate to do the fine-tuning.

#### 3.2 Training approach for the final educational chatbot

Our final model is based on the "cedpsam/chatbot\_fr" pre-trained conversational model. This is a "a dialoggpt model trained on french opensubtitles with custom tokenizer." [6]. We then fine-tuned this model with our data. The principle of the fine-tuning of this pre-trained model is presented in Fig.2.

The input is a batch of strings. These strings are in the following format "<startofstring> question Possible answers are: [choices]. <bot>: correct\_answer. explanation <endofstring>".

We apply the "cedpsam/chatbot\_fr" tokenizer to get the encoded input, and then we apply the "cedpsam/chatbot\_fr" model. We get the loss of the outputs and we backpropagate to do the fine-tuning.

### 4 Experiments

#### 4.1 Data

The first dataset come from the CS-552 EPFL course and contains questions from EPFL professors' courses. This "CS-552 dataset" consists of two files. The "interactions" file contains the answers generated by the students using ChatGPT. Each question is presented several times with different answers. Each entry in this file is composed as follows: a confidence level between 0 and 5, the solution ID, the interaction ID and the interaction itself in the form of a list of dictionaries: ['role': 'system', 'content': "text-system", 'role': 'user', 'content': 'text-user', 'role': 'assistant', 'content': 'text-assistant', 'role': 'user', 'content': 'text-user', 'role': 'assistant', 'content': 'text-assistant', etc.]. The "solutions" file contains the correct answers from the EPFL teachers' courses. Each question is present only once. Each entry is presented as follows: a question, a list of choices, an answer, an explanation and an ID solution. The list of choices and the explanation are optional.

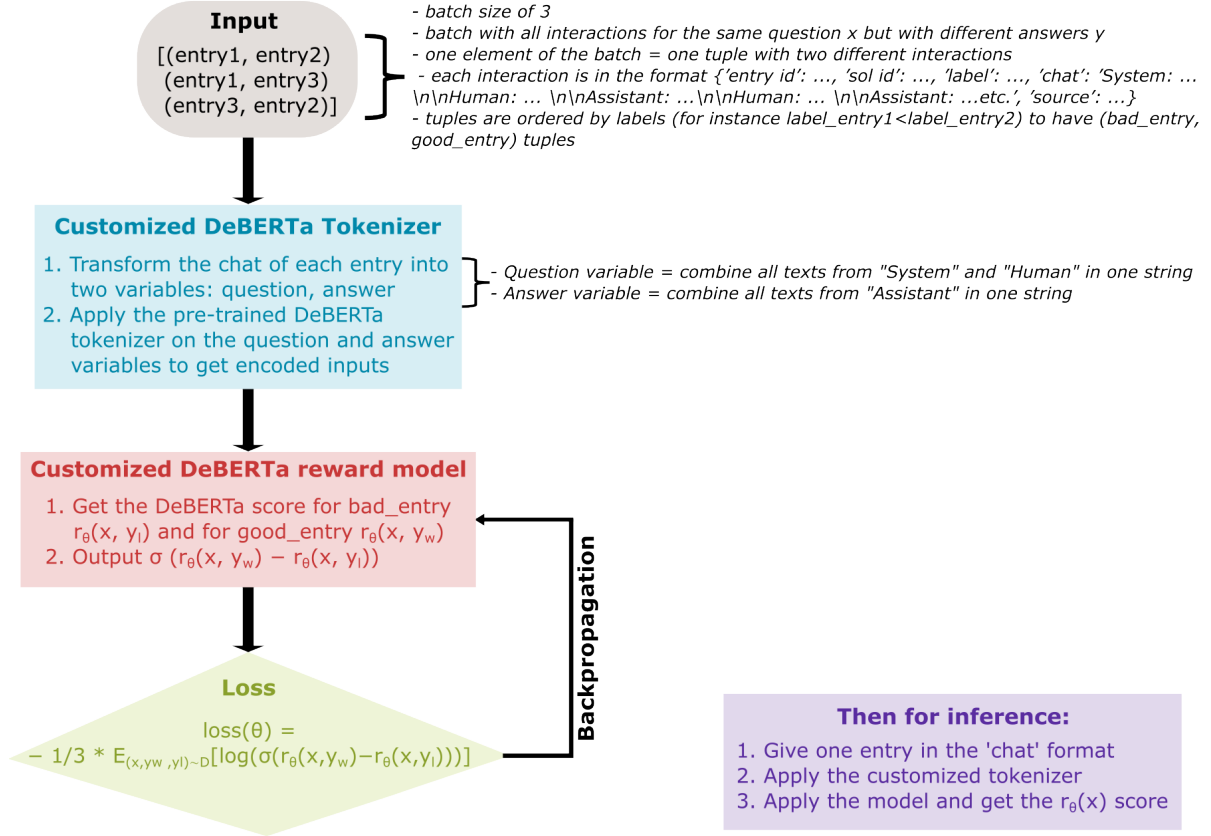


Figure 1: Principle of the fine-tuning of the reward model

The second dataset is the "SciQ dataset". The description of this dataset is as follows: "The SciQ dataset contains crowdsourced science exam questions about Physics, Chemistry and Biology, among others. The questions are in multiple-choice format with 4 answer options each. For the majority of the questions, an additional paragraph with supporting evidence for the correct answer is provided." [7]. Each entry in this dataset consists of a question, a correct answer, an explanation supporting the correct answer, and three incorrect answers. We will only use the test dataset, which contains 1,000 entries.

The third dataset is the "Exams dataset". The description of this dataset is as follows: "EXAMS is a benchmark dataset for multilingual and cross-lingual question answering from high school examinations. It consists of more than 24,000 high-quality high school exam questions in 16 languages, covering 8 language families and 24 school subjects from Natural Sciences and Social Sciences, among others." [8]. Specifically, we focused on the test dataset in French and with topics taught at EPFL.

For the fine-tuning of the reward model, we used the "CS-552 dataset" and the "SciQ dataset" only. Each entry of these datasets was put in the following format: {'entry\_id': 36644, 'sol\_id': 'sciq\_0', 'label': 0, 'chat': 'Human: question \n\nAssistant: answer. explanation (only for correct answer)', 'source': 'sciq dataset'}. The entry\_id is unique for each entry, the sol\_id is the same for interaction that correspond to the same question, the label is between 0 and 6 and informs about the quality of the answer, and the source indicate from which dataset is extracted the chat. The labels that come from the "interactions" file of the "CS-552 dataset" where given by hand by the students, the labels that come from the "solutions" file of the "CS-552 dataset" are always 6, and the labels that come from the "SciQ dataset" are 0 for the incorrect answers and 6 for the correct answer. Then for each given question, we kept only three interactions. The rules used for the filtering where the following for the "CSS-552 dataset": if the sol\_id was in the "solutions" file, we kept only the two worse entries in the "interactions" file with these sol\_id and the one in the "solutions" file, but



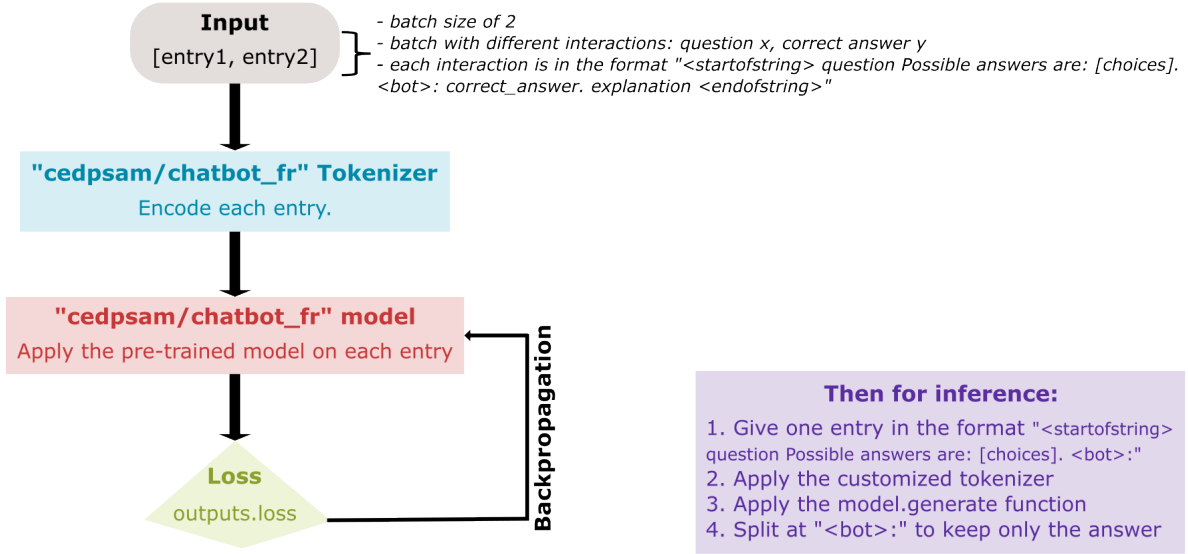


Figure 2: Principle of the fine-tuning of the final model

if the sol\_id was not in the "solutions" file, we kept the two worse entries and the best entry in the "interactions" file with these sol\_id. In the case of the "SciQ dataset", we kept each time the entry with the correct answer and two entries with incorrect answers. Then each dataset was split into train and test sets. The train CS-552 reward dataset had a size of 9444, and the test a size of 1050, and the train SciQ reward dataset had a size of 2700, and the test a size of 300. The datasets from each source were then merged into a final train and test datasets. Everything was shuffle and put in a batch of three, each batch containing the three combinations of the three interactions with the same sol\_id. The first interaction of each combinations was each time the worse one.

For the fine-tuning of the final educational model, we used the "solutions" file from the "CS-552 dataset", the correct answers from the "SciQ dataset" and the correct answers from the "Exams dataset". Each entry was put in the following format: "<startofstring> question Possible answers are: [choices]. <bot>: correct\_answer. explanation <endofstring>". Then each dataset was splitted into a train and a test set. The size of the train CS-552 generation dataset was 3655, the one of the test set was 407, the size of the train SciQ generation dataset was 900, the one of the test set was 100, and the size of the train Exams generation dataset was 229, the one of the test set was 26. The datasets from each source were then merged into a final train and test datasets. Everything was shuffle and

put in a batch of two.

## 4.2 Experimental details

To train the reward model, we used a batch size of 3, a max length of 500 in the tokenizer (the choice was made by splitting the input at spaces and looking at the distribution of the input length), a learning rate of 1e-4, the Adam optimizer, and the loss was  $\text{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x, y_w, y_l) \sim D} [\log(\sigma(r_\theta(x, y_w) - r_\theta(x, y_l)))]$ . The training was done with 2 epochs.

To train the final model, we used a batch size of 2, a max length of 400 in the tokenizer (again the choice was made by splitting the input at spaces and looking at the distribution of the input length), a learning rate of 1e-3, the Adam optimizer, and the loss was the loss of the pre-trained model directly. The training was done with 10 epochs.

## 4.3 Evaluation

To evaluate the reward model, we used our evaluation dataset. We computed the loss, we plotted the distribution of  $\sigma(r_\theta(x, y_w) - r_\theta(x, y_l))$ , and we looked at the percentage of times the model assigns a better score to the response we know to be better for the "CS-552 dataset". We also plotted the distribution of  $\sigma(r_\theta(x, y_w) - r_\theta(x, y_l))$  for all the data from the second epoch training.

To evaluate the final model, we also used our evaluation dataset. We computed the loss, we looked by hand at the interactions produced, we used the reward model, and we looked at the preci-

sion, the recall and the F1 score of the BERTscore. This evaluation scheme was repeated three times: once by evaluating the entire maximum likelihood decoded output, once by evaluating only the maximum likelihood decoded output after the "<bot>", and finally once by evaluating the output generated by the generate function (with a top\_p of 0.92 and a top\_k of 50).

#### 4.4 Results

The loss of the reward model remained around 0.11 throughout the training for both epochs. This suggests that the model has not really learned from the data and that our fine-tuning is not really adequate. A first very basic explanation could be an error in the code or simply in the loss implementation. In particular, we have to mention the fact that a major problem with our loss function at the moment is that we do not take into account the fact that the  $r_\theta$  score can be negative, which completely distorts the meaning of our actual loss. However, this error was noticed too late and so we'll have to start the training all over again to see if it really changes the results. A second explanation could be that our data is too diverse and the model has difficulty learning from it. The loss for the dataset evaluation was then 0.12. Fig.3 and Fig.4 show the distribution of  $\sigma(r_\theta(x, y_w) - r_\theta(x, y_l))$  for all the data from the second epoch training as well as for all the data from the dataset evaluation. We can see that the distribution is fairly uniform, with a larger peak at 1. If we then look only at the evaluation dataset data from the CS-552 dataset, we find that the model assigns a better score to the true solution than to the correct solution in 29% of cases.

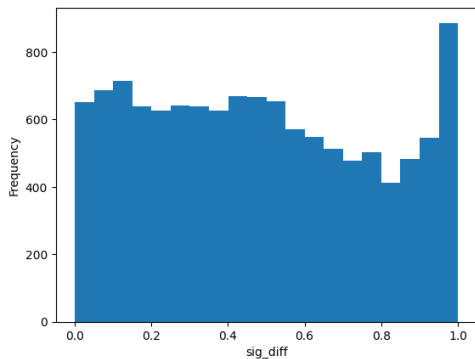


Figure 3: Histogram of  $\sigma(r_\theta(x, y_w) - r_\theta(x, y_l))$  for all the data from the second epoch training for the reward model

In the case of the final model, the loss decreased significantly during training, from 2.19 to 0.14. For

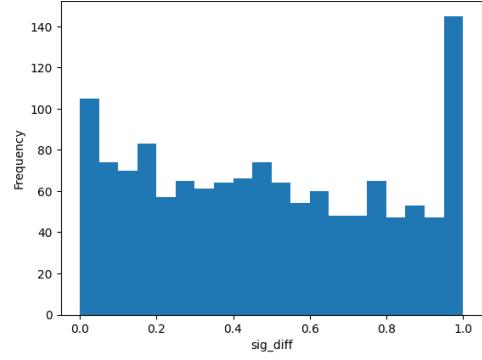


Figure 4: Histogram of  $\sigma(r_\theta(x, y_w) - r_\theta(x, y_l))$  for the evaluation data for the reward model

the evaluation, the most interesting is to start by looking at examples of the results directly. Table.1 shows examples with the question, the real solution and the generation by our model (using the generate function). We can see straight away that our model does not perform well at all. Not only does it fail to answer the question, but the sentences do not really make sense and often just repeat the same word. In view of these results, it does not really seem useful to show the results of the evaluation using the reward model and BERTscores, since it is obvious that our model does not perform well. A first explanation could be the base model. It was chosen because it allowed us to use interactions in English and French, but if we use this model directly without fine-tuning, we get even less consistent results. A second explanation could be the data, which may not be sufficiently numerous given the complexity of the task. For example, some data contains a huge amount of LaTeX symbols, which can add complexity. Another explanation is that the training/approach is not the most appropriate or sufficient here.

Although the results of the full evaluation are not shown here as they are not of particular interest in view of the performance of our model, all the code and graphics have been done and can be viewed on the GitHub page at the following link <https://github.com/CS-552/project-m3-rfl>.

## 5 Conclusion

Since the introduction of ChatGPT in November 2022, the landscape of educational chatbots has experienced a transformative revolution. The usage of this tool among students for their coursework has been consistently growing, achieving impressive records. However, leveraging chatbots for ed-

Question	Answer	Answer generated by the model
Vrai ou faux: pour savoir si la perméance propre d'un aimant ou d'une bobine varie en fonction de la position, on regarde si le flux créé par ce dernier varie en fonction de la position. Possible answers are: ['vrai', 'faux'].	vrai. Vrai, car la perméance est définie comme le flux divisé par le potentiel, donc si le flux varie en fonction de la position alors la perméance aussi.	ography est une bobine de claquage conçue en courant. AA. bf{J}}{D} de champ magnétique form de champ magnétique form de champ magnétique. ar sponges, le flux de courant est constant.
(Minima) Convex functions over a convex set have a unique global minimum. Possible answers are: ['True', 'False'].	False. False. Strictly convex functions have a unique global minimum, but this is not the case for convex functions in general.	ography matrixtake, where the non-intersecting closed intervals and the non-intersecting closed intervals are the same basic non-intersecting closed intervals. ween the concerne and the non-intersecting closed intervals.
Which of the following statements is true? Possible answers are: ['Is due to photoelectric absorption and Compton scatter.', 'Gives the relative number of x-rays emitted at each photon energy.', 'Depends only on the braking radiation.', 'Is produced by electron transitions between the electron shells.'].]	Gives the relative number of x-rays emitted at each photon energy.. The spectrum shows the superposition of characteristic X-ray lines and the Bremsstrahlung X-rays. Only characteristic X-ray are produced by electron transitions between the electron shells. Moreover, only the Bremsstrahlung X-rays are called "braking radiation".	gin the electron shells. Moreover, the green measure of "Optical electron transitions" and "Optical electron transitions".
Combien de molécules de NAD+ sont générées à partir d'une molécule de glucose par glycolyse ? Possible answers are: ['A une', 'B deux', 'C trois', 'D quatre', 'E zéro'].	E.	ography est une'ution de l'erieure, ici il y a desired getting de flux totaliséful (credit "A), avec fatty a un noyau de fertilike chez un noyau de fertilike [repeating "chez un noyau de fertilike" many times] d'un noyau d'un noyau d'un noyau d'hydrogène. C trois. jлизations, le noyau d'une quantité donnée d'une quantité donnée d'une quantité donnée d'une quantité donnée [repeating "d'une quantité donnée" many times]

Table 1: Examples of outputs of the model with the question, the real solution and the generation by our model (using the generate function)

educational purposes still poses several challenges, most notably the issue of ChatGPT generating answers that appear remarkably coherent but are factually incorrect. This challenge is particularly critical in the educational domain, where accuracy is paramount.

To tackle this challenge, we adopted the following approach. Firstly, we utilized ChatGPT to generate responses to questions derived from EPFL courses. For each question, we generated and evaluated three different interactions alongside the actual solution. We then employed these datasets and external datasets to fine-tune a reward model, aiming to assess the quality of chatbot responses, and to refine a conversational chatbot tailored for educational contexts. Although our results were disappointing, they represent an initial attempt to address the aforementioned challenge. The ultimate goal, which remains unexplored in this project, is to employ the reward model for Reinforcement Learning from human feedback (RLHF), which has the potential to bring substantial value to the field if further advancements are made.

Regarding our approach, it is evident that significant improvements are necessary. We need to continually scrutinize our methodology to gain a deeper understanding of the issues at hand and devise effective solutions. Several potential avenues for improvement have already been proposed, such as taking into account the negative sign of the score in the loss for the reward model, ensuring compatibility of datasets for training and considering normalization techniques, conducting more extensive training on larger datasets, and refining the methodology itself. Nevertheless, our journey has yielded substantial progress in our comprehending the field. Notably, although not detailed in this report, the training of the reward model underwent a complete overhaul after milestone 2 due to the inadequacy of the initial method for our specific data and model. Therefore, while the approach presented here is still unsatisfactory, it already represents a significant advance on our initial approaches to the problem and is the result of extensive refinement and investigation.

## 6 Team contributions

- **Félicie Giraud-Sauveur**: Everything
- **Romane Clerc & Louise Font**: Participation only in the writing of milestone1 outline

## References

- [1] *Large Language Models and Education - Maastricht University*. URL: <https://www.maastrichtuniversity.nl/large-language-models-and-education>.
- [2] Mohammad Amin Kuhail et al. “Interacting with educational chatbots: A systematic review”. en. In: *Education and Information Technologies* 28.1 (Jan. 2023), pp. 973–1018. ISSN: 1573-7608. DOI: [10.1007/s10639-022-11177-3](https://doi.org/10.1007/s10639-022-11177-3). URL: <https://doi.org/10.1007/s10639-022-11177-3>.
- [3] Chung Kwan Lo. “What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature”. en. In: *Education Sciences* 13.4 (Apr. 2023). Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, p. 410. ISSN: 2227-7102. DOI: [10.3390/educsci13040410](https://www.mdpi.com/2227-7102/13/4/410). URL: <https://www.mdpi.com/2227-7102/13/4/410>.
- [4] *OpenAssistant/reward-model-deberta-v3-large-v2* · Hugging Face. May 2023. URL: <https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2>.
- [5] Long Ouyang et al. *Training language models to follow instructions with human feedback*. arXiv:2203.02155 [cs]. Mar. 2022. DOI: [10.48550/arXiv.2203.02155](https://arxiv.org/abs/2203.02155). URL: <http://arxiv.org/abs/2203.02155>.
- [6] *cedpsam/chatbot\_fr* · Hugging Face. URL: [https://huggingface.co/cedpsam/chatbot\\_fr](https://huggingface.co/cedpsam/chatbot_fr).
- [7] *sciq* · Datasets at Hugging Face. June 2023. URL: <https://huggingface.co/datasets/sciq>.
- [8] *exams* · Datasets at Hugging Face. May 2023. URL: <https://huggingface.co/datasets/exams>.