

# Milestone 3, Chat-MMA

Antoine Magron, Mathieu Desponds, Mekhron Bobokhonov

## Abstract

Large Language Models (LLMs), such as ChatGPT, have largely democratized the use of natural language processing in mainstream use. However, the effectiveness of LLMs relies heavily on prompt engineering and instructions provided to the chatbot. This study focuses on developing a specialized chatbot tailored to answering EPFL course-related questions, eliminating the need for users to engage in prompt engineering. We employ advanced prompt engineering techniques on ChatGPT to generate a dataset of high-quality answers for EPFL course questions. Subsequently, we fine-tune a T5-small model using reinforcement learning from human feedback (RLHF) with the generated dataset. The paper provides a starting point for developing such chatbot for EPFL courses that can be improved by developing a bigger dataset and using more advanced models. By creating a dedicated chatbot and leveraging advanced prompt engineering techniques, our project aims to enhance the learning experience for EPFL students seeking course-related information.

## 1 Introduction

ChatGPT[7] represents a significant advancement in the field of Large Language Models (LLMs), allowing users to have access to these models. However, the performance of ChatGPT is heavily dependent on the quality of the prompt and instructions provided to the chatbot that can be hard to grasp for random users. Therefore, several studies have explored various techniques, such as few-shot prompting strategies and chain of thought, to enhance the prompt engineering process for LLMs.

The focus of our work is to develop a specialized chatbot tailored to answering questions related to EPFL courses, alleviating the need for users to worry about prompt engineering. Our goal is to offer students an effective platform for acquiring course-related information. By submitting questions to the chatbot, students can receive answers that include key theorems and lemmas relevant to answer the question. Additionally, the chatbot will provide explanations regarding the truth or falsehood of statements, offering

different pieces of information to support its response.

To achieve this, we employ prompt engineering techniques on ChatGPT to generate a dataset of high-quality answers for questions pertaining to EPFL courses that were provided by the different teacher of EPFL. Subsequently, we use this dataset to fine-tune a version of T-5[5], which is available online, using reinforcement learning from human feedback (RLHF[4]).

In this paper, we present a comprehensive exploration of the relevant literature in Section 2. We then delve into our approach to the problem in Section 3, describing the steps we took to develop the specialized chatbot and provide our experimental setup. In section 4, we continue with detailed analysis of the results in and finally conclude our work in Section 5, summarizing the contributions and discussing potential avenues for future research.

## 2 Related Work

### 2.1 Prompting

Large Language Models (LLMs) have demonstrated impressive performance and have become accessible to a wide range of users with the emergence of ChatGPT [7]. However, the effectiveness of these models heavily depends on the quality of the prompts provided to them.

Several techniques have been explored to enhance the prompting process. Brown et al. (2020) introduced the concept of few-shot prompting [1], which replaces traditional fine-tuning methods with large datasets, yielding promising results. Moving away from the task-specific few-shot paradigm that needs to be developed for each specific task, Reynolds et al. [6] proposed a different approach by designing metaprompts that are general prompts that can be easily adapted to specific tasks.

However, these techniques were not adapted for complex tasks. Trying to overcome this issue, the concept of Chain-of-Thought reasoning has been introduced by Wei et al. [8]. This technique involves introducing intermediate steps during the interaction with LLMs, leading to a more coherent and logical sequence of responses.

Recent research has questioned the effectiveness of manually crafted prompts, considering that LLMs fundamentally generate output word by word. Zhou et al. proposed an automatic prompt generation technique, which aims to find the optimal prompt for a given task based on a set of question-answer pairs.

## 2.2 Training

Training LLMs to build effective chatbots is challenging, as generating output aligned with user intent is not guaranteed. In 2022, Ouyang et al. [4] introduced a reinforcement learning technique based on human feedback to train chatbots, with the goal of better aligning their responses with user intent. This approach allows for training LLMs to generate responses in a desired format without the need for users to employ advanced prompt engineering techniques.

By exploring the literature on prompting techniques and training methodologies, this review highlights the importance of prompt design and introduces reinforcement learning from human feedback as a promising approach to enhance chatbot performance without heavy reliance on prompt engineering for the user.

## 3 Methods and experimental details

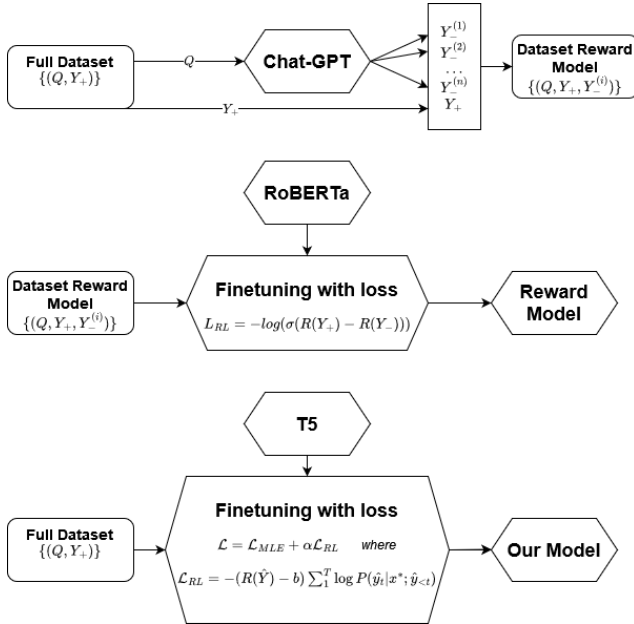


Figure 1: Project Plan

## 3.1 High-level methodology

The aim of the project was to build an assistant similar to ChatGPT to answer students' questions related to EPFL courses. The pipeline for building such an assistant consisted of three main steps:

- **Step 1: Collect demonstration data, and train a supervised policy.** We collected data by using questions from EPFL courses and employing different advanced prompting strategies to gather answers from ChatGPT for these questions. We used these answers as the desired output for our assistance. Subsequently, we fine-tuned a pretrained T5-small model on this data using supervised learning.
- **Step 2: Collect comparison data, and train a reward model.** We collected a dataset of comparisons between different answers to the same question. We then trained a reward model to predict the human-preferred output.
- **Step 3: Optimize a policy against the reward model using Reinforce: Basics.** We used the output of the reward model (RM) as a scalar reward. We fine-tuned the supervised policy to optimize this reward using the Reinforce:Basics technique, where we added an additional term to the usual maximum likelihood estimation (MLE) loss. The added term is defined as:

$$\mathcal{L}_{RL} = -(R(\hat{Y}) - b) \sum_1^T \log P(\hat{y}_t | x^*; \hat{y}_{<t})$$

where  $\hat{Y}$  is the generated response of the current state model to the given question  $x^*$ . The final loss is a combination of the MLE loss and the reinforcement learning (RL) loss:

$$\mathcal{L} = \mathcal{L}_{MLE} + \alpha \mathcal{L}_{RL}$$

where  $\alpha$  is a weight factor for balancing the two loss terms.

In this section we will describe each step in detail. Figure 1 contains the entire structure of the project.

## 3.2 Dataset

Our prompt dataset consisted of 4450 questions on EPFL courses. This dataset was split across more than 100 students. Each student employed effective prompt techniques

to engage ChatGPT and obtain answers to these questions. Our prompt strategies drew inspiration from the works of [6], [3], and [8], incorporating few-shot prompting with a chain of thought and assigning a role to the chatbot. The specific prompts used can be found in Appendix 6.1. This process yielded a dataset of 10,835 interactions, each assigned a confidence level indicating the model’s correctness in providing the answer. We filtered out answers with a confidence level below 3, those containing fewer than 6 sentences, or comprising fewer than 500 characters. These filtering choices struck a balance between retaining highly satisfactory answers and ensuring an adequate amount of data. Following this filtering process, we retained 6,068 samples. However, this resulted in 942 unanswered questions and 3,045 different questions for which we had lost some interactions. To address this, we decided to generate three answers using our three prompt strategies for the unanswered questions and regenerate one answer using our best prompt strategy for the remaining questions that had discarded samples. Consequently, our final dataset, used for the fine-tuning of our generative model, took the form of  $(Q, Y_+)$ , where  $Y_+$  represented well-generated answers, comprising a total of 10,948 samples.

### 3.2.1 Train data for Supervised fine-tuning

Since we didn’t have enough computational resources to train large models on a big corpus of data, we made the decision to focus only to the English language. As a result, for the supervised fine-tuning, we used only the questions on English from the mentioned above dataset. Additionally, for each question, we randomly selected and retained only one well-generated answer. After implementing this procedure, the training dataset consisted of 2882 pairs  $(Q, Y_+)$ , where  $Q$  represents the question and  $Y_+$  represents the well-generated answers and the evaluation dataset consisted of 321 questions.

### 3.2.2 Training data for the Reward model

The objective of the Reward Model was to assign high scores to good answers and low scores to bad answers. To train this model, we constructed a dataset consisting of tuples  $(Q, Y_+, Y_-)$ , where  $Q$  denoted a question,  $Y_+$  represented a good answer, and  $Y_-$  represented a bad answer. The bad answers ( $Y_-$ ) were generated using three baseline prompts, which can be found in Appendix 6.2, resulting in one bad answer per question. By combining the generation

of  $Y_+$  and  $Y_-$ , we obtained a total of 10,936 samples. Finally, we divided the dataset into three portions: a training dataset, accounting for 70% of the dataset; an evaluation dataset, constituting 15%; and a test dataset, comprising the remaining 15%.

### 3.2.3 Training data for the RL model

To train the Reinforce:Basic, we used the same dataset as for the SFT model.

## 3.3 Models

### 3.3.1 Supervised fine-tuning (SFT)

We began by using the T5 pre-trained language model. T5 is an encoder-decoder model that has been pre-trained on a mixture of unsupervised and supervised tasks, with each task converted into a text-to-text format. T5 shows good performance on various tasks by incorporating a specific prefix for each task in the input. For instance, for translation, the prefix would be "translate English to German: ..." and for summarization, it would be "summarize: ...".

Due to the limitations imposed by computational resources, we decided to use the smaller version of the T5 model, which contains 60 million parameters. We performed fine-tuning on T5-small using supervised learning with our labeled demonstrations. As input to the model, we used prompts in the following format:

Answer to the question: {question}

Answer options: {choices}

The target for the model was the well-generated ChatGPT answers that were collected.

Like all autoregressive language models, the model was trained to generate the next word of the target response  $y_t^*$  based on a set of preceding words in the target  $y_{<t}^*$  and the instruction  $x^*$ . The loss function used was the negative log-likelihood of the target response.

$$\mathcal{L}_{MLE} = - \sum_1^T \log P(y_t^* | y_{<t}^*; x^*)$$

The training process consisted of 10 epochs, with a batch size of 4 and a learning rate of  $2e - 5$ .

### 3.3.2 Reward Model

We began by training a reward model that is supposed to take an interaction  $Y$  as input and return  $R(Y)$ , a real number that will represent how good the demonstration is. The model will yield a high reward for a response that has the structure of the "good" response we queried through chat-GPT and a low reward for the "bad" ones. Consider a tuple  $(Q, Y_+, Y_-)$  where  $Y_-$  is a "bad" answer and  $Y_+$  is a "good" answer to  $Q$ , the goal of our model is to maximize the reward between them which lead to the following loss function :

$$\mathcal{L}_{RL} = -\log(\sigma(R(Y_+)) - \sigma(R(Y_-)))$$

The reward model is composed of two submodels:

- A Bidirectional Transformer Encoder. We feed it the tokenized preprocessed input sentence and we keep the embedding of a prepended [CLS]. This embedding contains information on the entire sequence of interaction.
- A Feed Forward Network that takes the sentence embedding, and outputs a single number. The dimensions of the hidden layers and the activations functions used in the model are modular and different combinations will be tested.

After a few tests, we chose RoBERTa-base for the language model and a Feed Forward Network with 3 hidden layers. The input layer is the size of RoBERTa's hidden dimension, the three hidden layers have sizes 128, 64, and 16 respectively. The activation function between each of these layers is ReLU. We then finally output a single number and apply no activation function to be able to have a greater range of reward function modeling.

To assess the quality of our models we will use three metrics :

- The average difference in reward between associated  $Y_+$  and  $Y_-$
- The accuracy in the ordering i.e. the proportion of pairs  $(Y_+, Y_-)$  for which  $R(Y_+) > R(Y_-)$

The model will, later on, be used to assign a reward to an output of a blank generative model. That is why we used regression and not classification. We will favor models that span a large range of value to account for the large amount of possible answers given by the generative model.

### 3.3.3 Final Generative Model

This section corresponds to the bottom of Figure 1. In this part, we performed fine-tuning of the SFT model to optimize the output of the reward model. To achieve this, we incorporated the basic Reinforce loss in addition to the usual maximum likelihood estimation (MLE) loss for generative language models. During the training the model provided by batch of prompts of questions like for SFT model and target responses for this question:  $(Q, Y_+)$ . Then the following steps were performed during training:

- Calculation of the negative log likelihood of the target outputs:

$$\mathcal{L}_{MLE} = -\sum_1^T \log P(y_t^* | y_{<t}^*; x^*)$$

- For each prompt from the batch, we generated a response using the current state of the model. Then, using the reward model, we estimated the reward for the generated answers. Finally, we calculated the RL-part loss:

$$\mathcal{L}_{RL} = -(R(\hat{Y}) - b) \sum_1^T \log P(\hat{y}_t | x^*; \hat{y}_{<t})$$

where  $\hat{Y}$  is the generated response of the current state model to the given prompt  $x^*$ ,  $R(\hat{Y})$  is the output of the reward model, and  $b$  is a constant baseline. The RL-part loss encourages the model to increase the probability of the sampled token in the same context, proportional to the reward value.

- The final loss is a combination of the MLE loss and the RL loss:

$$\mathcal{L} = \mathcal{L}_{MLE} + \alpha \mathcal{L}_{RL}$$

where  $\alpha$  is a weight factor for balancing the two loss terms.

We trained our final model using the described loss function. We experimented with different values of  $\alpha$ , and the best performance was observed with  $\alpha = 0.1$ . In Figure 2, shown the distribution of reward model values. We can observe that for bad answers, it gives negative values. However, in order for the optimization of the RL loss to be reasonable, the rewards should be positive. Therefore, we chose  $b = -5$  as a baseline.

In general, the hyperparameter values used were:  $\alpha = 0.1$ ,  $b = -5$ ,  $\text{num\_epochs} = 4$ ,  $\text{batch\_size} = 2$ ,  $\text{learning\_rate} = 2e - 5$ .

### 3.3.4 Baseline

As the baseline model, we used our SFT-model. To evaluate the performance of our final model, we compared it to the SFT model using the Reward model. Additionally, we assessed the quality of the generated answers from both models.

## 4 Results

### 4.1 Reward Model

All the Reward Models were trained with an AdamW optimizer with a warmup phase of 30%.

We began by building a baseline model with the same structure by training on a smaller set of samples (approximately a quarter of the total samples) for only one epoch. This model will, expectantly yield the worst performances.

We, then, trained a second model on the same subset of the sample for 5 epochs. With the training sample being still small we don't expect the best performances either.

Finally, we trained a model on the complete set of training samples for 5 epochs.

Each training was done on Google Cloud Platform using GPUs to speed up the process. The following metrics were obtained :

	Correct ranking (%)	Average Difference
Baseline (barely trained)	15%	$-3.69 \times 10^{-9}$
model_hf_97 (undertrained)	66%	0.18
model_hf_full (fully trained model)	94.4%	5.49

Table 1: Metrics of the reward models

We can see that the correct ranking percentage grows with the training, starting at 15% on a barely trained model to reach 94.4%. After an investigation, we realized that the model's misclassifications were on ambiguous samples where both  $Y_-$  and  $Y_+$  were good, or  $Y_-$  was even better than  $Y_+$ . You can see in Figure 2 the distribution of the

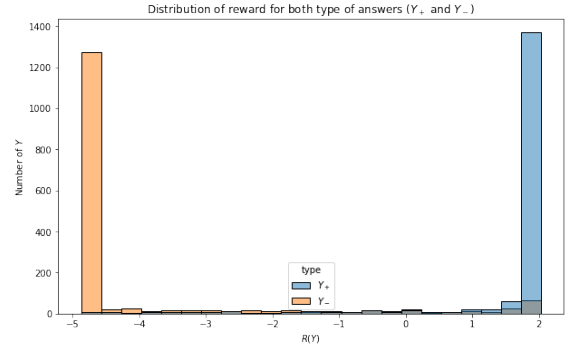


Figure 2: Distribution of  $R(Y)$

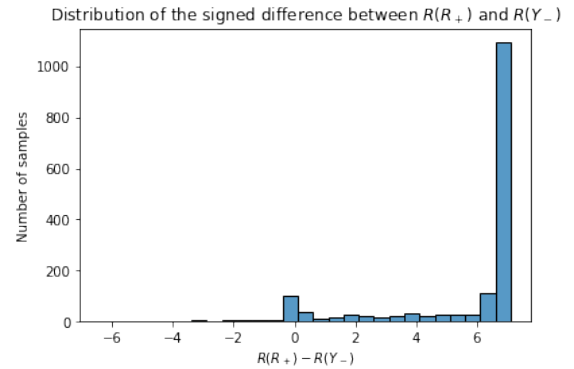


Figure 3: Distribution of the signed difference between associated  $R(Y_+)$  and  $R(Y_-)$

rewards for both "good" and "bad" answers. The values of  $R(Y)$  span a wide range of values with two clear peaks for strong classification. This is the expected behavior of a good reward model. Figure 3 shows the distribution in difference  $R(Y_+) - R(Y_-)$  for associated "bad" and "good" answers.

When doing reinforcement learning we also need to compute a baseline to account for the "expected reward". By subtracting the baseline, we can focus on the deviations from the expected reward, which can provide more informative signals for learning. The simple solution for this is to take the average reward,  $-1.24$ .

### 4.2 Final Model (RLHF)

In this section, we will present the results of our final model, compare it with the base model, and evaluate the average reward value assigned to the generated responses by the reward model.

### 4.2.1 Qualitative results

We experimented with various decoding strategies, including greedy decoding, beam search, top-k sampling, and top-p sampling. Through our evaluation, we found that the beam search decoding strategy yielded the best results. Greedy decoding sometimes resulted in repetitive sentences.

For the final generation of responses, we employed the following parameters for beam search decoding: num\_beams = 5, no\_repeat\_ngram\_size = 2, early\_stopping = True, max\_length=128.

Let's examine some examples of generated answers from both the SFT and RLHF models:

*Example 1.*

#### **Question:**

Would you describe this learning procedure as reinforcement, supervised or unsupervised learning?

#### **Answer options:**

0) It is supervised, since we explicitly provide the correct weights to initialize the network.

1) It is unsupervised since the network learns implicit associations present in the input without any additional teaching signal.

2) It is reinforcement learning since only weight updates only occur when a pattern is retrieved correctly.

#### **SFT generated answer:**

The question asks about this learning procedure as reinforcement, supervised or unsupervised learning. Answer: 0: "It's not the same thing as a physical learning technique, since we explicitly explicitly provide the correct weights to initialize the network. This is because we explicitly provides the right to Initialize": False. The answer is false. Therefore, the answer to the question is true

#### **RLHF generated answer:**

The question asks about this learning procedure as reinforcement, supervised or unsupervised learning. The answer answer is 0: It is the reinforcement learning, since we explicitly provide the correct weights to initialize the network. Answer 2: "Employment

learning": False. This means that a network learns implicit associations present in the input without any additional teaching signal. Therefore, the answer option is not correct.

*Example 2.*

#### **Question:**

Which resolution is improved with CT over projection radiography?

#### **Answer options:**

0) Temporal resolution

1) Energy resolution

2) Contrast resolution

3) Spatial resolution

#### **SFT generated answer:**

The question asks about which resolution is improved with CT over projection radiography. This answer answer is 0: Temporal resolution (CMR) - Contrast resolution, and spatial resolution. It is also a common answer to the question asking about the CT resolution that can be improved by CT.

#### **RLHF generated answer:**

The question asks about which resolution is improved with CT over projection radiography. Answer: 0: Temporal resolution (energy resolution) 3) Contrast resolution. The answer: "Energie-resolution" : False. It is not a resolution, but it does not improve the resolution of CT. Therefore, the answer is incorrect. This is true, since CT is an effective resolution that can be used to reduce the intensity of the radiation.

*Example 3.*

#### **Question:**

In the lecture on bias-variance decomposition we have seen that the true error can be decomposed into noise, bias and variance terms. What happens to the three terms for ridge regression when the regularization parameter  $\lambda$  grows? Explain your answer.

#### **SFT generated answer:**

The question asks about the true error can be decomposed into noise, bias and variance terms. In the lecture on bias-variance-decomposition, we have seen that the false error is possible



in noise. This is because the correct answer is: “The correct answers” are the three terms for ridge regression when the regularization parameter  $\lambda$  grows. It is not a true answer, but it does not necessarily mean that it will not be true. Therefore, it is true that there is no real answer to this question.

#### RLHF generated answer:

The question asks about the lecture on bias-variance decomposition. This is because the true error can be decomposed into noise, bias and variance terms. The correct answer is:  $\lambda$  is a type of ridge regression variable that is used for regression regression. Using the same algorithm, we can use bias terms to determine the correct error. Therefore, if the regularization parameter increases the value of the variable value, it is less likely to increase the number of variables that are used in regression analysis.

#### 4.2.2 Discussion of the Results

When comparing the models, both models initially restate the question in their answers. However, the SFT-generated answer lacks clarity and fails to provide a clear explanation or reasoning for the chosen option. It also contains repetitive and confusing language, making it difficult to grasp the intended meaning.

In contrast, the answer generated by the RLHF model is clearer and more coherent compared to the SFT-generated answer. It directly addresses the question and provides an explanation for the selected option. However, it still does not provide the correct answer to the question. This discrepancy may be attributed to the use of a less powerful pre-trained model. Due to computational limitations, we utilized T5-small. If we had employed a larger version of T5, our RLHF implementation might have been capable of producing correct answers to the questions.

Overall, the answers generated by the RLHF model demonstrate improved quality compared to those generated by the SFT model. They exhibit clearer patterns, address the questions more directly, and offer more accurate evaluations of the answer options. The RLHF model displays a better understanding of the questions and generates more coherent responses, making it a more reliable choice for answer generation. However, further improvements are still

needed, and utilizing a more powerful model, increasing the number of epochs, or enlarging the dataset used for fine-tuning would be beneficial.

#### 4.2.3 Evaluation using a Reward Model

Using the SFT model and RLHF model, we generated answers for the provided testing questions in `prompts.json` and calculated the average reward value of the Reward Model for the generated answers.

Model	Sample size	Average rewards	Range of possible rewards
SFT	100	1.419	[-5; 2]
RLHF	100	1.826	[-5; 2]

Table 1: Average reward for generated responses

With this evaluation method, the RLHF model once again demonstrates better performance compared to the SFT model. Figure 2 displays the distribution of reward model values, indicating that the range of Reward Model outputs spans from -5 to 2. Notably, the answers generated by the RLHF model have an average reward value of 1.82, which is relatively high.

### 5 Discussion

Our study aimed to develop a specialized chatbot tailored to answering questions related to EPFL courses, eliminating the need for users to engage in prompt engineering. By employing advanced prompt engineering techniques on ChatGPT and fine-tuning a T5-small model using reinforcement learning from human feedback (RLHF), we have made significant progress towards achieving this goal.

Our results demonstrate that the RLHF model outperforms the SFT model in generating answers for EPFL course-related questions. The answers generated by the RLHF model exhibit clearer patterns, address the questions more directly, and provide more accurate evaluations of the answer options. However, it is important to note that the RLHF model still falls short of providing the correct answer to the questions in some cases. This limitation could be attributed to the use of a less powerful pre-trained model, T5-small, due to computational constraints.

To improve the performance of our chatbot, several steps can be taken in future work. First, employing more powerful pre-trained language models, such as `Flan-T5` [2], could

lead to better results. Flan-T5, trained on a much larger dataset of 800 million parameters, has demonstrated superior performance and understanding of natural language tasks. Leveraging such advanced models could enhance the chatbot’s ability to provide accurate and informative answers to EPFL course questions.

Furthermore, increasing the training time and running more epochs during fine-tuning could contribute to the refinement of the chatbot’s responses. Our study used a limited number of epochs due to computational and time limitations, and extending this duration could potentially improve the model’s understanding of the intricacies of EPFL course-related inquiries. Additionally, expanding the dataset used for fine-tuning would provide the model with a wider range of high-quality answers, leading to more accurate responses.

In conclusion, our work presents a promising starting point for developing a chatbot dedicated to answering EPFL course-related questions without the need for prompt engineering. The combination of advanced prompt engineering techniques, RLHF, and the utilization of T5-small has resulted in improved performance compared to a baseline model. However, there is still room for improvement, and future research should focus on utilizing more powerful models, increasing training time and dataset size, exploring additional training techniques, and refining the reward model. These advancements would contribute to the development of a more accurate and reliable chatbot, enhancing the learning experience for EPFL students seeking course-related information.

## References

- [1] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: [2005.14165 \[cs.CL\]](#).
- [2] Hyung Won Chung et al. *Scaling Instruction-Finetuned Language Models*. 2022. arXiv: [2210.11416 \[cs.LG\]](#).
- [3] Andrew K. Lampinen et al. *Can language models learn from explanations in context?* 2022. arXiv: [2204.02329 \[cs.CL\]](#).
- [4] Long Ouyang et al. *Training language models to follow instructions with human feedback*. 2022. arXiv: [2203.02155 \[cs.CL\]](#).
- [5] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020. arXiv: [1910.10683 \[cs.LG\]](#).
- [6] Laria Reynolds and Kyle McDonell. *Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm*. 2021. arXiv: [2102.07350 \[cs.CL\]](#).
- [7] Kadhim Hayawi Sakib Shahriar. “Let’s have a chat! A Conversation with ChatGPT: Technology, Applications, and Limitations”. In: (2023). URL: <https://arxiv.org/abs/2302.13817>.
- [8] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: [2201.11903 \[cs.CL\]](#).



## 6 Appendix

### 6.1 Prompting strategies for $Y_+$ generation

#### 6.1.1 1<sup>st</sup> prompting strategy

##### Prompt

Here is the question : `sample['question']`.

Here are the answers : `sample['answers']`.  
There is a multiple choice question with `len(sample['choices'])` possible solutions.

Here is the question : `sample['question']`.

Here are the answers : `sample['answers']`.

**Instruction** You are a university teacher for Imaging course. You gave homework to your students and now correct it. For each question, only suppose what is written in the question.

You first explain what is asked in the question. Then you give definitions, theorems and lemmas that will be usefull to answer the question.

Next, develop each possible answers saying if it is true or false and why it is the case by giving an different argument for each answers. For each answer structure it by using the following pattern "Answer #x : <the xth answer> : <True or False>. The reason why you answered this

An answer is true if we can not find a counter example. An answer is false if we can find a counter example that show that the statement is false. If the answer is false provide a counter-example. Finally give the number of the final answer (the is unique except if in the question it is said that there might be multiple answers) that is true in any cases.

#### 6.1.2 2<sup>nd</sup> prompting strategy

**Prompt** You are taking an exam for a university course. You will be given a multiple-choice question and your task is to select only one correct answer from the offered list of options.

To solve the problem, follow these steps:

- Explain what are the important informations needed in order to answer correctly the

problem like theorems, general information about terms in the question and formulas.

- Go step by step through all possible answers from the given list of options.
- For each option, try to think whether this answer is correct or not and explain your reasoning. Is this a complete and appropriate answer to the question asked?
- When you have checked all the answers from the given list of options, choose the most complete and appropriate answer among them. Do not choose the correct answer until you have checked all the answers from the given list.

Please, for each option from the offered list write an explanation of why you think it's false or true.

Use the following format: Question: question here

Choices: here is a list of possible answers

Important informations: Here are the important informations needed in order to answer correctly the problem.

Correct Answer: Here is the correct answer that you chose from the offered list. format: variant number. answer

Explanation: Here is an explanation and justification for each option why it's true or false. format: variant number. "TRUE" or "FALSE": Your explanation

Question: `sample['question']`

Choices: `sample['choices']`

#### 6.1.3 3<sup>rd</sup> prompting strategy

**Instruction** You are a teacher in an Institute of technology. Your goal is to provide correct and well explained answer so that your student can learn from your explanation.

Start by stating the definitions, lemmas, theorems, and formulas that are relevant to the resolution of the exercise.

You'll have a question and multiple proposed answers. You will list all the answers in the

order they were proposed. For each of them you will propose a reasoning justified by definitions and properties

In a multiple choice question, a proposition is considered true when we can't find any counter example. It has to be true all the time.

You will finish by simply stating what the answer was.

Here's examples of interactions :

Q : There is a multiple choice question with 4 possible solutions. Here is the question : What is the stationary distribution of a Markov chain?. Here is the answers : 1. "The distribution of the initial state of the chain" 2. "The distribution of the final state of the chain" 3. "The distribution that the chain converges to as the number of steps approaches infinity" 4. "The distribution that the chain starts with and maintains throughout its evolution"

A : To understand the concept of stationary distribution of a Markov chain, we need to first understand what a Markov chain is. A Markov chain is a mathematical model that describes a system that transitions between different states over time. The transition from one state to another is based only on the current state of the system and not on any previous states.

The stationary distribution of a Markov chain is the distribution that the chain converges to as the number of steps approaches infinity. This means that no matter what the initial state of the chain is, after a large number of steps, the probability of being in each state will approach a fixed distribution, which is the stationary distribution.

1. "The distribution of the initial state of the chain": False. The distribution of the initial state of the chain is simply the probability distribution of starting in each state. It does not take into account the transitions between states.

2. "The distribution of the final state of the chain": False. The distribution of the final state of the chain depends on the initial

state and the transition probabilities. It is not fixed and does not represent the stationary distribution.

3. "The distribution that the chain converges to as the number of steps approaches infinity": True. As explained above, the stationary distribution is the distribution that the chain converges to as the number of steps approaches infinity.

4. "The distribution that the chain starts with and maintains throughout its evolution": Partially True. The stationary distribution is the distribution that the chain approaches after a large number of steps. It is nevertheless possible for a Markov chain to start at the stationary distribution but it's not the usual case.

Therefore, the correct answer is 3. "The distribution that the chain converges to as the number of steps approaches infinity".

Q : There is a multiple choice question with 4 possible solutions. Here is the question : What is the voltage drop through a resistor of  $1M\Omega$  of a current with intensity 15 A. Here is the answers : 1. "10V" 2. "100 000 V" 3. "10 000 000 V" 4. "15 000 000 V" A : A resistor is an electrical component that limits the flow of current in a circuit. The voltage drop across a resistor is given by Ohm's law:  $V = IR$ , where  $V$  is the voltage drop,  $I$  is the current, and  $R$  is the resistance.

In this case, the resistor has a resistance of  $1M\Omega$  and the current has an intensity of 15 A. Therefore, the voltage drop across the resistor is:

$$V = IR = 15 \text{ A} * 1M\Omega = 15,000,000 \text{ V}$$

Answer 4, "15,000,000 V" is the correct answer.

Q : There is a multiple choice question with 4 possible solutions. Here is the question : What is the Fourier transform of

$$f(x) = e^{-\alpha x} u(t)$$

. Here is the answers : 1.

$$\frac{\beta}{\alpha + i\omega}$$

2.

$$\frac{\alpha}{\beta + i\omega}$$

3.

$$\frac{\alpha}{\beta - i\omega}$$

4. "The fourier transform doesn't exist" A : To find the Fourier transform of the given function, we can start by applying the definition of Fourier transform, which is:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$

Here, we have  $f(t) = \beta e^{-\alpha t} u(t)$ , where  $\beta$  and  $\alpha$  are constants, and  $u(t)$  is the unit step function.

First, we can simplify  $f(t)$  using the fact that  $u(t) = 0$  for  $t < 0$  and  $u(t) = 1$  for  $t \geq 0$ . Then, we have:

$$f(t) = \beta e^{-\alpha t} u(t) = \begin{cases} 0, & t < 0 \\ \beta e^{-\alpha t}, & t \geq 0 \end{cases}$$

Now, we can substitute this expression into the definition of Fourier transform and solve the integral. We have:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt = \int_0^{\infty} \beta e^{-\alpha t} e^{-i\omega t} dt$$

We can simplify this integral by factoring out  $\beta$  and using the fact that  $e^{a+b} = e^a e^b$ . Then, we have:

$$F(\omega) = \beta \int_0^{\infty} e^{-(\alpha+i\omega)t} dt = \beta \left[ \frac{-1}{\alpha+i\omega} e^{-(\alpha+i\omega)t} \right]_0^{\infty}$$

Since  $e^{-\infty} = 0$ , we can simplify this expression as:

$$F(\omega) = \frac{\beta}{\alpha + i\omega}$$

Therefore, the correct answer is 1.  $\frac{\beta}{\alpha + i\omega}$ .

Q : There is a multiple choice question with 4 possible solutions. Here is the question : What is the mathematical equation for the Fourier transform of a continuous signal?. Here is the answers : 1. "

$$F(u) = \int f(t)e^{-2\pi i u t} dt$$

" 2. "

$$F(t) = \int f(u)e^{2\pi i t u} du$$

" 3. "

$$F(u) = \sum f(t)e^{2\pi i u t}$$

" 4. "

$$F(t) = \sum f(u)e^{2\pi i t u}$$

" A : To answer this question, we need to know the mathematical equation for the Fourier transform of a continuous signal. The Fourier transform is a mathematical tool used to decompose a signal into its frequency components.

1. "

$$F(u) = \int f(t)e^{-2\pi i u t} dt$$

" : True. This is the correct mathematical equation for the Fourier transform of a continuous signal. It represents the integral of the signal multiplied by a complex exponential.

2. "

$$F(t) = \int f(u)e^{2\pi i t u} du$$

" : False. This equation is not the correct mathematical equation for the Fourier transform of a continuous signal. The variable "t" and "u" are swapped, and the complex exponential has the opposite sign in the exponent.

3. "

$$F(u) = \sum f(t)e^{2\pi i u t}$$

" : False. This equation represents the Fourier series of a periodic signal, not the Fourier transform of a continuous signal. The "sum" symbol implies that the signal is periodic and can be decomposed into a series of complex exponentials.

4. "

$$F(t) = \sum f(u)e^{2\pi i t u}$$

" : False. This equation is not the correct mathematical equation for the Fourier transform of a continuous signal. The variable "t" and "u" are swapped, and the complex exponential has the opposite sign in the exponent.

Therefore, the correct answer is 1. "

$$F(u) = \int f(t)e^{-2\pi i u t} dt$$

".

Q : There is a multiple choice question with 4 possible solutions. Here is the question : You have two continuous time signals that are of exponential order , how can you compare their internal frequencies ?. Here is the answers : 1. "the signals known values" 2. "the Z transform" 3. "their derivatives" 4. "their Laplace transform" A : The two studied signals are unbounded. In signal processing, an unbounded signal is a signal that grows without limit as time increases, or a signal that has an infinite energy or power. They also are said to be of exponential order. A signal  $f(t)$  is said to be of exponential order if there exist positive constants  $M$  and  $a$  such that  $|f(t)| \leq Me^{at}$  for all  $t \geq 0$ . In other words, the signal grows no faster than an exponential function as  $t$  increases. To compare the internal frequencies of two continuous time signals that are of exponential order, we need to understand the properties of these signals and the mathematical tools that can be used to analyze them.

1. "The signals known values": False. The values of the signals alone do not provide enough information to compare their internal frequencies. The signals could have different initial values or amplitudes, but still have the same internal frequency.

2. "The Z transform": False. The Z transform is a mathematical tool used to analyze discrete time signals, not continuous time signals. It is not applicable in this case.

3. "Their derivatives": False. The derivatives of the signals can provide information about the rate of change of the signals, but they do not directly provide information about their internal frequencies.

4. "Their Laplace transform": True. The Laplace transform is a mathematical tool used to analyze continuous time signals, and it can provide information about the internal frequencies of exponential order signals. The Laplace transform of an exponential order signal is a rational function, and the poles of this

function correspond to the internal frequencies of the signal.

## 6.2 Prompting strategies for $Y_-$ generation

### 6.2.1 For Multiple Choice Questions

1. **instruction:** Develop each possible solution saying if they can be true or false and why it is the case. Finally give the number of the final and unique answer that is true in any cases.

**prompt:** There is a multiple choice question with `len('choices')` possible solutions. Here is the question : 'question'. Here are the answers : 'answers'

2. **prompt:** You are taking an exam for a university course. You will be given a multiple-choice question and your task is to select only one correct answer from the offered list of options. Use the following format: Question: "" question here "" Choices: "" here is a list of possible answers "" Correct Answer: "" Here is the correct answer that you chose from the offered list. format: variant number. answer ""

Explanation: "" Here is an explanation and justification for each option why it's true or false. format: variant number. "TRUE" or "FALSE": Your explanation ""

3. **instruction:** You a teacher in an insitute of technology. Your goal is to answer student's question and help them understand the answer. It needs to be very clear. You will list the proposition and whether they are true or false. You will end by stating the right answer among them and justify this one is correct.

**prompt:** Here is a multiple choice question with `len('choices')` possible solutions. Here is the question : 'question'. Here are the choices : 'answers' ""

### 6.2.2 For Simple Questions

1. **instruction:** Answer to the question and say why it is your answer

**prompt:** 'question'

2. **prompt:** You are taking an exam for a university course. You will be given a question and your task is to answer correctly to the given question. Use the following format: Question: “ question here “  
Correct Answer: “ your answer here “  
Explanation: “ here is an explanation and justification of your answer. “

3. **instruction:** The same instruction as in 3 prompt for MCQ.

**prompt:** Here is the question : 'question'.