DD2417 - Project Report

Sebastien Roig (`roig@kth.se`) **Alexander Gutell** (`agutell@kth.se`)

May 2024

Abstract

In this work we tried to fine-tune large language models on a summarization task on the wikihow dataset [KW18]. More precisely, we made our experimentations with Mistral-7B and LLama-2-B. We used different parameter efficient tuning techniques such as QLoRA [Det+23] to be able to run the computations on limited resources. We proceeded to a hyper-parameter comparison on the low-rank adapter parameters, and used the best model for our final tuning. We found that descent results could be achieved, even with few steps of training, and with a low adapter rank. Our code is available at `https://github.com/ByTaizer/Project_LLM`.

1 Introduction

Summarization is an important task in natural language processing which aims to reduce the text as concisely as possible, bringing out the main ideas. The two methods that stand out are extractive summarization, where the main ideas are selected from the text and assembled, and abstractive summarization, where the summary is written from scratch. Abstractive summarization has recently become a more popular solution with the emergence of large language models (LLM) that allows state of the art performances on all natural language processing tasks. In particular, pre-trained Sequence-to-sequence models such as [Dev+19] have been direct candidates for fine-tuning for this kind of task. Moreover, with the development of parameter efficient tuning (PEFT) techniques such as LoRA [Hu+21] or QLoRA[Det+23], the accessibility of these language models has been greatly improved. In this work, we investigate the capabilities of the recent large language models such as Mistral-7B [Jia+23] or LLAMA-2-7B in this context. The fine-tuning is done on the dataset wikihow [KW18] and the evaluation is done through the ROUGE metric [Lin04] on another part of the dataset.

2 Related work

To fine-tune the pre-trained models we had to go through several previously known concepts.

2.1 Decoder Transformers

The models we use are pre-trained Transformers. More precisely, as we wanted to use the most state of the art models, we focused on decoder-only models. The most common architecture for text summarization is Sequence-to-sequence architecture which are encoder-decoders as proposed in the original Transformer paper [Vas+23], due to its straightforward representation of the training task (input, label). The decoder-only architecture (Causal Language modeling) on the contrary, is not straightforward to fine-tune for this task, because one has to think about ways to make the model learn to predict in a certain format. Thus, what we wanted to learn is how to fine-tune these models through well thought training prompts.

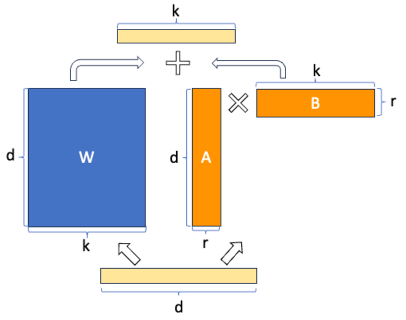
2.2 LoRA

Low-rank adaptation [Hu+21] is a method developed to reduce the amount of trainable parameters while maintaining really good results after fine-tuning. It has previously been shown in some papers ([AZG20]) that the space in which operates the learned over-parametrized models lies in a lower intrinsic dimension. What is hypothesized in this paper is the fact that the change in the model weights during fine-tuning would also lie in a lower dimension. What it means is that one could train lower rank adaptations to the model weights. Here, it is done in each layer by adding a low-rank update matrix during training and freezing the original ones. If we call the original frozen weights of a specific layer $W_0 \in \mathbb{R}^{d \times k}$ and ΔW the update matrix, ΔW is constrained by a low rank decomposition being $\Delta W = BA$ with $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ and $r << \min(d, k)$. The forward pass is then expressed as follows :

$$Wx = W_0 x + \frac{\alpha}{r} \Delta W x,$$

with α a scaling factor that is roughly equivalent to tune as the learning rate of the model. As we want to initialize those upgrades at zero, A is initialized as a Gaussian initialization and B with zeros.

Figure 1: LoRA adapters weight matrices.



2.3 Quantization

Quantization is a technique that allows to store the model weights in a lower memory consuming representation. A lot of research work has been done to quantize large models while trying to maintain their original capacities during inference. Recent work with QLoRA [Det+23], extended the previous works by allowing the training as well to be done in the quantized setup. QLoRA is the combination of the two techniques presented before. A schematic representation of the memory usage during training can be seen in 2. The main ideas is that the model weights are stored in a special format (4-bit Normal-Float) that allows to reduce as much as possible the loss of information, and the training is done only on low-rank adapters in BFloat-16 precision thanks to a process of dequantization during the computation of the gradients. In reality, there are more technical details that can be found in the original paper. In our case, this allows us to store the 7 billion parameters models on only 5 GB of GPU memory whereas before approximately 16 GB was necessary. For the training, we could fit the model with a batch size of 4 whereas it was not even doable to train it with a batch size of 1.

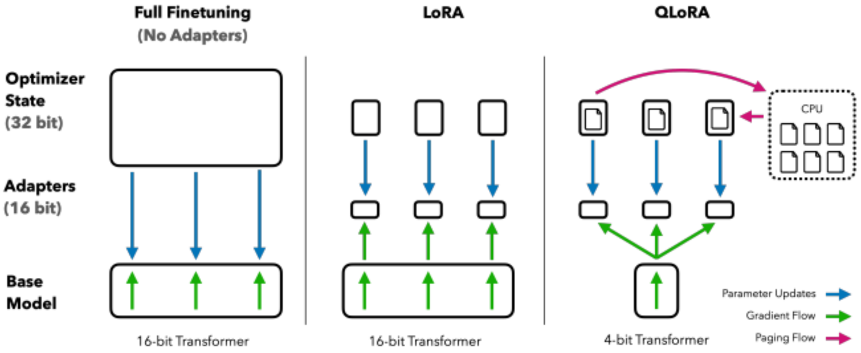


Figure 2: Different fine-tuning methods and their memory requirements.

3 Dataset

The file wikihowSep.csv contains data from the online knowledge base WikiHow. The data was extracted as described in the original paper, see [KW18]. The file consists of four headers:

- Title: Article title.
- Overview: Article overview.

- Headline: Segment headlines (excluding title and overview).
- Text: Content under each segment.

Where the headlines represent the summaries and the text is the body of summarization.

4 Software and hardware

All code was written in python where the following modules were utilized: `huggingface_hub`, `datasets`, `transformers`, `accelerate`, `peft`, `bitsandbytes` and `wandb`.

Transformers is a library from huggingface that provides APIs to download and train state of the art models [Fac24c]. The base class Dataset implements a Dataset backed by an Apache Arrow table, which is used by the transformers library. [Fac24b]. PEFT (parameter efficient tuning) simplifies fine tuning by freezing weights and adding trainable adapters to the model. Bitsandbytes [Fac24a] enables 4-bit quantization and training with QLoRA [Det+23].

This project is being carried out on the Jupyter Hub instance provided for this course on a 10 GB H100 GPU.

5 Methods

5.1 Overview

The three models that were evaluated were Llama-2-7B, Mistral-7B and opt-350m [Zha+22]. The methodology between the two models is roughly the same and was conducted in parallel by each of us individually. Although we agreed on the majority of methodology aspects, others were less well implemented. We are aware of the difference in results that this can lead to, particularly in the final comparison of performance between the models, but because of the time limit, we have decided to keep the results obtained, bearing in mind that this should not drastically change the results.

5.2 Data Preprocessing

The data was first filtered in the following ways. For Mistral, all sentences (text + headline) above length 512 was removed, this was due to the limitation of RAM. For Llama-2-7B we proceeded another way and chose to remove the data where the length was outside of one standard deviation $\mu \pm \sigma$. The aim here was to retrieve a more homogeneous data set, to decrease the complexity, and therefore induce some regularization in the training process. We formatted the data in a specific way to invoke a summarization behavior (prompt engineering) by the model. The prompt formatting for the training data can be seen below.

Llama-2-7B:

```
prompt = "###Summarize: text ###Summary: headline </s>"
```

Mistral-7B:

```
prompt = "[START]Summarize: text Summary: headline [END]"
```

After the text had been formatted, the data was tokenized by the respective tokenizers belonging to the models. The data was then considered ready for training. At inference, for the Mistral-7B model, a stop-string argument is used to stop when the token '[END]' is encountered.

5.3 Incremental Approach and Experimentation

Fine-tuning LLMs is a computationally and time demanding task. In order to simplify the process and make experimentation less daunting, we began with a smaller model called opt-350m from Meta. Our aim was to find a general hypothesis on how to fine-tune an LLM for our specific purpose.

5.4 Opt-350m

Opt-350m is a 350 million parameters causal model that allowed us to familiarize ourselves with decoder-only models and how to train them. We did our first experimentation on this model with LoRA adapters with different hyper-parameter values to see how it influenced the training results. The model was trained on 10000 rows of the dataset for 600 steps and one of the results can be seen in 4. What we observed is that the model has difficulties converging to a clear summarizer behavior. We also observe that the model has a tendency to loop, which is something frequent for decoder-only models. What we would investigate for a further study would be to see if the model can achieve similar results as the two other models if trained on the whole dataset.

5.5 Training Mistral and Llama-2-7B

We started by adding adapters to all linear layers of the models as recommended in [Det+23], with the LoRA parameter alpha set to $\alpha = 16$ which is the default value. In the LoRA paper it is implied that we should fix this value as a constant, and then vary the rank r as the hyper parameter [Hu+21]. The overall hyper-parameters we assessed during this phase were: r, batch size, learning rate and optimization steps. After experimentation, we hypothesized that a learning rate of $2 \cdot 10^{-4}$ with a low number of optimization steps was sufficient to achieve good results. The batch size was set to 4 as a compromise between good results and less amount or RAM used. We trained for 100 steps with a gradient accumulation set to 4, i.e. 1600 data points were trained on. The results showed promising, which led us to decide upon tuning the rank parameter. There are many choices that can be made regarding the hyper parameters, but since both time and computational resources were limited, a greedy approach seemed reasonable.

Five training runs per model, with the specified parameters, were conducted for $r = \{1, 4, 8, 16, 32\}$. To see more details about the precise training scheme and parameters, check out the code.

5.6 Evaluation

Each version of the models (one for each r) was evaluated on 1000 data points with the ROUGE metric. The following scheme was used:

1. sample 1000 data points and partition them into batches of 100 samples.
2. compute the average ROUGE score over each batch.
3. calculate the standard mean and standard deviation over the batch results.

5.7 Further Analysis

After having found a satisfying rank order for our two competing models we proceeded to a final train on 500 steps on the best model between Mistral-7B and Llama-2-7B to see how it would improve the results and to compare its performances to referenced baselines [KW18]. The other hyper-parameters remain the same as previous experiment. For its evaluation, we used Rouge score again on the same test set.

6 Results

6.1 Llama2 7B

The evaluation loss for the respective ranks r was very similar as can be seen in fig.3b. The same outcome is reflected in the ROUGE results, which can be found in tab.1. Although rank 32 scored the best results, there is not a big difference in performance between the different ranks. A more subjective result is displayed with some example prompts and summaries in the appendix, see fig.6.

Rank order r	Rouge1	Rouge2	RougeL
1	28.83 ± 2.37	14.00 ± 2.85	28.26 ± 2.40
4	29.00 ± 2.38	13.92 ± 2.26	28.50 ± 2.39
8	29.39 ± 2.30	14.19 ± 2.29	28.74 ± 2.28
16	29.12 ± 2.27	14.19 ± 2.11	28.49 ± 2.22
32	29.42 ± 2.45	14.41 ± 2.01	28.82 ± 2.41

Table 1: Rouge scores with different rank orders for Llama-2-7B on a 100 steps training with 100 data points for testing. The evaluation is repeated 10 times to compute a standard deviation.

6.2 Mistral-7B

As for the Llama-2-7B model we trained the model for five different values of rank r on 100 epochs. The results 2 show a really good summarization capabilities even for a rank order $r = 1$, which confirms the hypothesis from LoRA paper. The evaluation is repeated 10 times on 100 different sampled data points. The final value that we chose for the final experiment is $r = 16$ as it is the one performing the best on most metrics. This choice is obviously not totally rigorous as we would have needed a longer training time, and a larger testing set. Indeed as we can see on figure 3a, the next points of the training process are uncertain and we can't really know which model would perform the best.

Rank order r	Rouge1	Rouge2	RougeL
1	34.43 ± 2.85	17.28 ± 3.35	33.41 ± 2.83
4	34.53 ± 3.01	17.24 ± 2.97	33.62 ± 2.94
8	34.46 ± 2.88	16.79 ± 3.17	33.56 ± 2.85
16	34.90 ± 2.51	17.44 ± 2.76	34.14 ± 2.59
32	34.91 ± 2.47	17.42 ± 2.97	33.99 ± 2.49

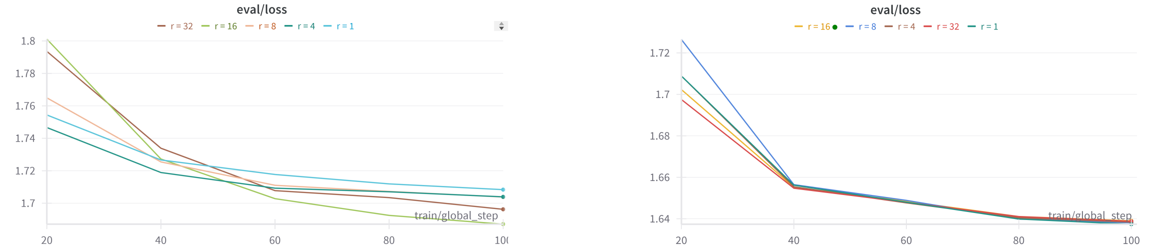
Table 2: Rouge scores with different rank orders for Mistral-7B on a 100 steps training with 100 data points for testing. The evaluation is repeated 10 times to compute a standard deviation.

6.3 Final results

For this final step, we chose Mistral-7B with rank $r = 16$ to train it for 500 steps. We proceeded similarly as in previous sections. We repeat 10 times the Rouge score computation on 100 sampled test samples. In 3 we see that the model has really good performances near the referenced state of the art (SOTA) on this dataset. Obviously, we cannot really compare our results to this model as our evaluation is only done a really small portion of the whole dataset.

7 Conclusion and Limitations

We saw in this work that causal LLMs can have really convincing summarization capabilities after only a few steps of training. We have been able to carry out fine-tuning of state of the art 7 billion parameters models on



(a) Rank order training comparison on 100 steps with Mistral-7B

(b) Rank order training comparison on 100 steps with Llama-2-7B

Figure 3: Comparison of rank order training on 100 steps

Rank order r	Rouge1	Rouge2	RougeL
Mistral-7B	35.76 ± 2.32	18.33 ± 2.46	33.98 ± 2.49
BertSum (SOTA)	35.91	13.9	34.82

Table 3: Rouge scores comparison for Mistral-7B trained on 500 steps with 100 data points for testing. The evaluation is repeated 10 times to compute a standard deviation. The BertSum scores come from the original paper [Liu19]

a quite limited amount of GPU resources thanks to recent parameter efficient tuning techniques. Moreover, the possibility to achieve this type of result with low rank adapters e.g. $r = 1$, not only confirmed the results from the LoRA paper [Hu+21], but also implies that this characteristic can be maintained in quantized models. The fact that the rank don't seem to have a big impact, also complies with the authors results in the QLoRA paper [Det+23]. Since a lower rank means less GPU needed to train the model, it might be the best option to start the fine tuning with $r = 1$ when working with limited resources.

The first limitation of the methodology comes from the dataset, which only contains ”how to” summaries. Thus, the model only learns to predict such summary formats, and doesn't really generalize to other type of summaries. It cannot be considered as a general summarizer but rather is a good model on this specific dataset. Something else to note about the dataset is that it contains examples where the summary contains more information than in the original text 5. Some future work would be to use a more general dataset or a mix of different types of datasets. Secondly, due to memory constraints, we limited the maximum number of tokens per example to 512 tokens and thus the model would not be able to summarize general texts such as books or other long texts. Another limitation comes from the time and resource constraints which did not allow us to rigorously perform hyper-parameter tuning for LoRA for example or cross-validation in the choice of the model. Finally, we performed the training and the evaluation on a really limited amount of data (10000 samples for training, 1000 samples for evaluation) whereas the original dataset is really large ($>$ 1 million rows), it would be interesting to see the results on the whole dataset for future work.

References

[AZG20] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. *Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning*. 2020. arXiv: 2012.13255 [cs.LG].

[Det+23] Tim Dettmers et al. *QLoRA: Efficient Finetuning of Quantized LLMs*. arXiv:2305.14314 [cs]. May 2023. DOI: 10.48550/arXiv.2305.14314. URL: http://arxiv.org/abs/2305.14314 (visited on 05/21/2024).

[Dev+19] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv:1810.04805 [cs]. May 2019. DOI: 10.48550/arXiv.1810.04805. URL: http://arxiv.org/abs/1810.04805 (visited on 05/21/2024).

[Fac24a] Hugging Face. *Bitsandbytes Documentation*. https://huggingface.co/docs/bitsandbytes/main/en/index. Accessed: 2024-05-21. 2024.

[Fac24b] Hugging Face. *Datasets Documentation*. Accessed: 2024-05-20. 2024. URL: https://huggingface.co/docs/datasets/v2.19.0/en/package_reference/main_classes#datasets.Dataset.

[Fac24c] Hugging Face. *Transformers Documentation*. Accessed: 2024-05-20. 2024. URL: https://huggingface.co/docs/transformers/main/en/index.

[Hu+21] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv:2106.09685 [cs]. Oct. 2021. DOI: 10.48550/arXiv.2106.09685. URL: http://arxiv.org/abs/2106.09685 (visited on 05/21/2024).

[Jia+23] Albert Q. Jiang et al. *Mistral 7B*. arXiv:2310.06825 [cs]. Oct. 2023. DOI: 10.48550/arXiv.2310.06825. URL: http://arxiv.org/abs/2310.06825 (visited on 05/21/2024).

[KW18] Mahnaz Koupaee and William Yang Wang. "WikiHow: A Large Scale Text Summarization Dataset". In: *CoRR* abs/1810.09305 (2018). arXiv: 1810.09305. URL: http://arxiv.org/abs/1810.09305.

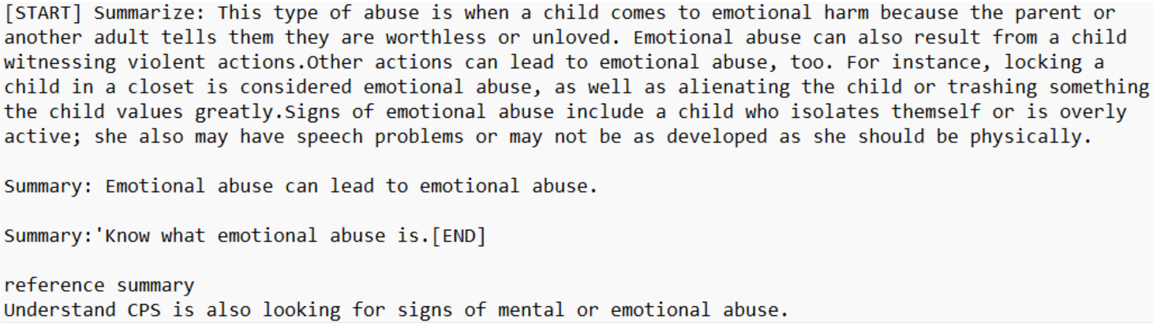
[Lin04] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[Liu19] Yang Liu. *Fine-tune BERT for Extractive Summarization*. 2019. arXiv: 1903.10318 [cs.CL].

[Vas+23] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

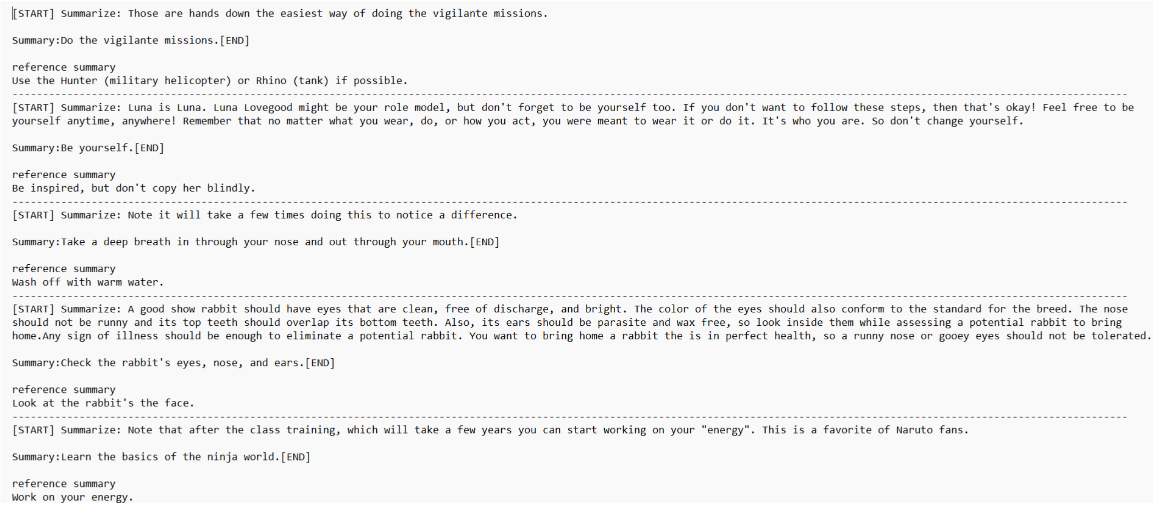
[Zha+22] Susan Zhang et al. *OPT: Open Pre-trained Transformer Language Models*. arXiv:2205.01068 [cs]. June 2022. DOI: 10.48550/arXiv.2205.01068. URL: http://arxiv.org/abs/2205.01068 (visited on 05/21/2024).

Appendix



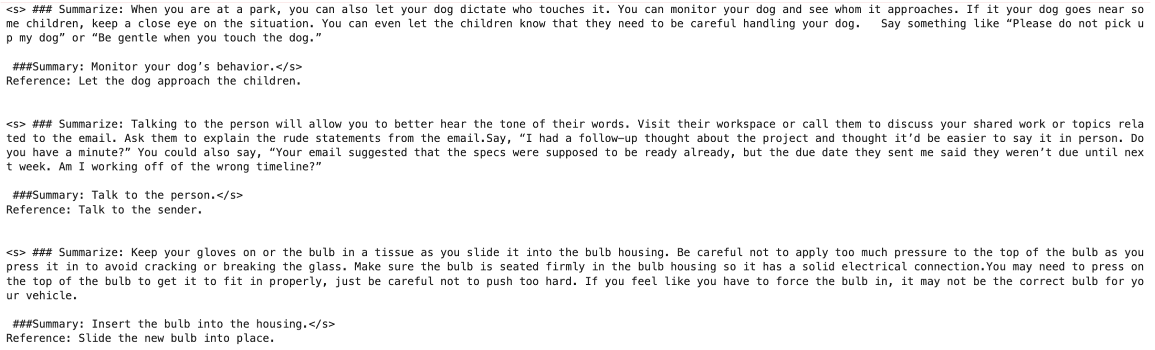
```
[START] Summarize: This type of abuse is when a child comes to emotional harm because the parent or
another adult tells them they are worthless or unloved. Emotional abuse can also result from a child
witnessing violent actions.Other actions can lead to emotional abuse, too. For instance, locking a
child in a closet is considered emotional abuse, as well as alienating the child or trashing something
the child values greatly.Signs of emotional abuse include a child who isolates themself or is overly
active; she also may have speech problems or may not be as developed as she should be physically.

Summary: Emotional abuse can lead to emotional abuse.

Summary:'Know what emotional abuse is.[END]

reference summary
Understand CPS is also looking for signs of mental or emotional abuse.
```

Figure 4: Summary example for opt-350m model, obtained with learning rate of $2 \cdot 10^{-4}$, $\alpha = 16$, $r = 8$, and the same formatting as Mistral fine-tuning



```
[START] Summarize: Those are hands down the easiest way of doing the vigilante missions.

Summary:Do the vigilante missions.[END]

reference summary
Use the Hunter (military helicopter) or Rhino (tank) if possible.
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
[START] Summarize: Luna is Luna. Luna Lovegood might be your role model, but don't forget to be yourself too. If you don't want to follow these steps, then that's okay! Feel free to be
yourself anytime, anywhere! Remember that no matter what you wear, do, or how you act, you were meant to wear it or do it. It's who you are. So don't change yourself.

Summary:Be yourself.[END]

reference summary
Be inspired, but don't copy her blindly.
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
[START] Summarize: Note it will take a few times doing this to notice a difference.

Summary:Take a deep breath in through your nose and out through your mouth.[END]

reference summary
Wash off with warm water.
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
[START] Summarize: A good show rabbit should have eyes that are clean, free of discharge, and bright. The color of the eyes should also conform to the standard for the breed. The nose
should not be runny and its top teeth should overlap its bottom teeth. Also, its ears should be parasite and wax free, so look inside them while assessing a potential rabbit to bring
home.Any sign of illness should be enough to eliminate a potential rabbit. You want to bring home a rabbit the is in perfect health, so a runny nose or gooey eyes should not be tolerated.

Summary:Check the rabbit's eyes, nose, and ears.[END]

reference summary
Look at the rabbit's the face.
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
[START] Summarize: Note that after the class training, which will take a few years you can start working on your "energy". This is a favorite of Naruto fans.

Summary:Learn the basics of the ninja world.[END]

reference summary
Work on your energy.
```

Figure 5: Summary examples with the final Mistral-7B



```
<s> ### Summarize: When you are at a park, you can also let your dog dictate who touches it. You can monitor your dog and see whom it approaches. If it your dog goes near so
me children, keep a close eye on the situation. You can even let the children know that they need to be careful handling your dog.   Say something like “Please do not pick u
p my dog” or “Be gentle when you touch the dog.”

 ###Summary: Monitor your dog’s behavior.</s>
Reference: Let the dog approach the children.


<s> ### Summarize: Talking to the person will allow you to better hear the tone of their words. Visit their workspace or call them to discuss your shared work or topics rela
ted to the email. Ask them to explain the rude statements from the email.Say, “I had a follow–up thought about the project and thought it’d be easier to say it in person. Do
you have a minute?” You could also say, “Your email suggested that the specs were supposed to be ready already, but the due date they sent me said they weren’t due until nex
t week. Am I working off of the wrong timeline?”

 ###Summary: Talk to the person.</s>
Reference: Talk to the sender.


<s> ### Summarize: Keep your gloves on or the bulb in a tissue as you slide it into the bulb housing. Be careful not to apply too much pressure to the top of the bulb as you
press it in to avoid cracking or breaking the glass. Make sure the bulb is seated firmly in the bulb housing so it has a solid electrical connection.You may need to press on
the top of the bulb to get it to fit in properly, just be careful not to push too hard. If you feel like you have to force the bulb in, it may not be the correct bulb for yo
ur vehicle.

 ###Summary: Insert the bulb into the housing.</s>
Reference: Slide the new bulb into place.
```

Figure 6: Summary examples with the rank $r = 32$ Llama-2-7B