Mathias Perez, Theo Lependeven
Team : 6

March 14, 2025

PROJECT: FINE-TUNE A SMALL LANGUAGE MODEL (SLM) FOR SUMMARIZATION

CONTENTS

INTRODUCTION

The objective of this project is to fine-tune a Small Language Model (SLM) for summarization, focusing on non-English texts. By leveraging synthetic summaries generated by a larger language model, our goal is to adapt an SLM (with less than 7B parameters) to produce concise and informative summaries. This project addresses challenges such as limited computational resources (Google Colab free-tier GPU with 16GB memory), dataset annotation in a non-English language, and effective evaluation against established metrics.

The code for our project is fully available on our repository: `https://github.com/Trick5t3r/Finetune-LLM-CASS.git`.

1 THE DATASET

For this project, we employed the CASS dataset, created by Bouscarrat et al. in 2019, which is available at `https://metatext.io/datasets/cass`. This dataset is composed of decisions rendered by the French Court of Cassation, accompanied by summaries crafted by lawyers. With 129,445 documents stored, the CASS dataset offers a rich and professionally annotated resource in French.

The expert summaries provided in the dataset will serve as a robust standard, allowing us to perform rigorous evaluations of our summarization model by comparing the generated outputs against these carefully curated references.

In order to augment the dataset for fine-tuning our Small Language Model (SLM), we generated additional synthetic summaries. To achieve this, we utilized the "bigscience/bloom-7b1" model. The generation process was guided by a carefully designed prompt engineering prefix to ensure the summaries were detailed and comprehensive.

During the summary generation process, several hyperparameters were meticulously tuned to balance diversity and relevance. Key settings such as the temperature, maximum token length, and top-k sampling were selected based on preliminary experiments, ensuring that the synthetic summaries maintained high quality and aligned well with the style and detail level of the expert summaries.

Our initial step involved evaluating the summaries produced by our high-capacity model to determine whether the synthetic dataset met our quality standards. By comparing these generated summaries against established benchmarks, we ensured that the data we intended to use for fine-tuning was robust and reliable.

Below is a table outlining various benchmark metrics that we considered during the evaluation process. Currently, the evaluation scores are marked as "na" (not available), pending further analysis:

Table 1: Benchmark Evaluation Metrics for Synthetic Summaries

Metric	ROUGE-L	BLEU	External model evaluation (GPT3,5)
Mean Evaluation Score	0.31	0.16	8,5/10

Subsequent evaluations involved further prompt-engineering, fine-tuning and hyperparameter adjustments aimed at enhancing the overall quality of the generated summaries. For ease of tracking performance changes across different configurations, we adopted a color-coded system[1].

[1] green indicates improvements in summary quality (e.g., better coherence, completeness, and contextual integration), while red denotes a decline in performance.

2
METHODOLOGY

2.1 Model Architectures

In our project, we explored two distinct model architectures for the summarization task: an autoregressive model and a sequence-to-sequence (seq2seq) model. The autoregressive approach was implemented using the Bloom Small 560M model, which generates text token by token in a unidirectional manner. In contrast, the seq2seq model we used was based on T5 [3], employing an encoder-decoder framework. This architecture processes the entire input sequence with an encoder to capture a global context before the decoder generates the output, allowing for more coherent summarization. The difference between these architectures lies primarily in how they handle context, with seq2seq models generally excelling in tasks where understanding the whole input is crucial.

2.2 Prompt Engineering, LoRA Optimization, and Fine-Tuning

To enhance the performance of our models, we combined three key techniques: prompt engineering, Low-Rank Adaptation (LoRA), and full fine-tuning.

Prompt Engineering We refined our input prompts to ensure that the models generated structured and coherent summaries. This step was crucial in guiding the models towards producing relevant outputs.

LoRA Optimization To optimize training efficiency, we applied LoRA, a parameter-efficient fine-tuning method. LoRA allowed us to introduce low-rank modifications to specific layers of our models, significantly reducing computational costs while maintaining performance.

Full Fine-Tuning In addition to LoRA, we performed full fine-tuning on both models to adapt them to our summarization task. This involved updating all model parameters on our curated dataset, which included both synthetic and professionally crafted summaries from the CASS dataset.

2.3 Fine-tuning Strategy and Hyperparameter Exploration

For both model types, we conducted extensive fine-tuning experiments to optimize performance. Our strategy involved:

- **Training:** Fine-tuning both the Bloom Small 560M and T5 models using our dataset. We applied both full fine-tuning and LoRA-based adaptations, depending on the computational constraints and performance trade-offs.
- **Hyperparameter Tuning:** Experimenting with various hyperparameters, such as the number of epochs, learning rates, and temperature settings during sampling. These experiments helped us balance output diversity and relevance effectively.

2.4 Evaluation Strategy

After fine-tuning, we evaluated the performance of both models by comparing their generated summaries with two reference standards: the synthetic summaries created by our high-capacity model and the professional summaries from the CASS dataset. This dual evaluation approach provided a comprehensive view of each model's performance and helped identify the optimal fine-tuning configuration.

For evaluation metrics, we employed standard measures widely used in the summarization literature. In particular, we computed the ROUGE-L, BLEU, and BERTScore metrics. The ROUGE-L score captures the longest common subsequence between the generated summary and the reference, while the BLEU score evaluates n-gram overlap. Additionally, BERTScore leverages contextual embeddings from a pre-trained BERT model to compute similarity between generated and reference summaries, providing a more nuanced evaluation of semantic accuracy. These scores were determined in comparison with both the original summaries and the summaries generated by our high-capacity model.

In summary, our methodology combined the strengths of different model architectures with targeted prompt engineering, efficient training via LoRA, and thorough hyperparameter exploration, enabling us to systematically compare and optimize summarization performance.

3
RESULTS

3.1 Autoregressive Models

3.1.1 • Training Task: Next-Token Prediction

For the autoregressive approach, our training task was formulated as a next-token prediction problem. This means that the model was trained to predict the subsequent token in a sequence, given all preceding tokens. Although this approach is standard for language generation, it posed challenges for our summarization task, where capturing the overall context is essential for producing a coherent and complete summary.

3.1.2 • LoRA Fine-Tuning for Autoregressive Models

To enhance performance, we applied Low-Rank Adaptation (LoRA) during fine-tuning of the Bloom Small 560M model which allowed us to train from 1% to 10 % of our parameters. LoRA was intended to streamline the training process by reducing the number of trainable parameters, thereby enabling a more efficient adaptation of the pretrained model to our task. Despite these efforts, the fine-tuning process did not yield the desired improvements. The fine-tuned model still produced outputs that were characterized by incoherent speech patterns and incomplete rephrasing of the source texts.

3.1.3 • Evaluation and Observations

In our evaluation, we observed that the pretrained autoregressive model, even without fine-tuning, generated relatively coherent speech but produced poor summaries that lacked completeness and failed to capture essential details. With LoRA fine-tuning, the model's performance did not improve significantly; in fact, the generated summaries remained incomplete and often fragmented. This outcome underscores a known limitation of autoregressive models in fully integrating context for tasks that require summarization. Given the clear shortcomings identified through human analysis, we decided not to pursue further quantitative benchmarking for this approach. These observations align with the findings in [1], which highlight that encoder-decoder architectures maintain better focus on the input than decoder-only models for conditioned generation tasks.

3.2 Seq2Seq Model 1 : T5

In this section, we present our first sequence-to-sequence model: T5. T5, introduced by Raffel et al. [3], is a unified text-to-text transformer model that reframes all NLP tasks as text generation problems. Pretrained on the massive C4 dataset—which consists primarily of English language text—the model is designed to handle a wide range of tasks, from translation to summarization and question answering. Despite its strong performance on English tasks, its application to French summarization required additional adaptations. In our work, we retrained T5-base (with 223M parameters) on our augmented dataset to better capture the nuances of French legal texts in a summarization task.

3.2.1 • Initial Evaluation of the Pretrained T5-Base Model

For the sequence-to-sequence (seq2seq) approach, we began by evaluating the performance of our T5-base pretrained model without any additional fine-tuning. This preliminary evaluation served as a baseline to understand the model's inherent capabilities in summarization. The initial results indicated that, while the model was able to grasp the task concept, it struggled to execute it effectively.

The primary reason for this shortfall is that the pretrained T5-base model was originally trained on English data. Without further training, it was unable to transfer its summarization capabilities to French texts. In contrast, when the model is tested on English documents, it produces significantly better summaries.

Table 2: Benchmark Evaluation Metrics for the Pretrained T5 Model against Reference Summaries

Metric	ROUGE-L	BLEU	External Model Evaluation (GPT 3.5)[1]
Pretrained Model Score	0.0753	0.0019	1,5/10

[1] GPT-3.5 evaluation seems unreliable as it can be biased by the context provided by the user.

3.3 Seq2Seq Model 2 : mT5

In this section, we present our second sequence-to-sequence model: mT5. mT5 (with 580M parameters) is the multilingual variant of the T5 architecture, pretrained on the mC4 dataset—a multilingual extension of the C4 corpus. It is important to note that mT5 was exclusively pre-trained in an unsupervised manner on mC4, without any supervised training on downstream tasks. Consequently, the model requires fine-tuning before it can be effectively applied to specific tasks, such as summarization. However, due to its multilingual design, mT5 appears particularly well-suited for handling non-English texts, including French legal documents.

3.3.1 • Initial Evaluation of the Pretrained mT5-Base Model

Again, we began by evaluating the pretrained mT5-base model without any additional fine-tuning. Although mT5 is pretrained on the multilingual mC4 dataset, its unsupervised training did not include task-specific adjustments for summarization. As a result, when applied to French texts, the model struggled to generate coherent and complete summaries. In contrast, its performance on English documents was noticeably better, likely due to a closer alignment with the distribution of its pretraining data.

Table 3: Benchmark Evaluation Metrics for the Pretrained mT5 Model against Reference Summaries

Metric	ROUGE-L	BLEU	External Model Evaluation (GPT 3.5)
Pretrained Model Score	0.0863	0.0135	1/10

3.4 Prompt Engineering

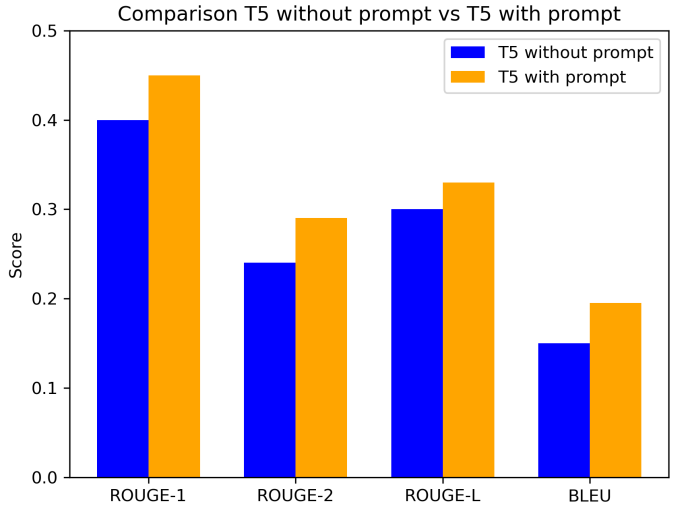
Our prompt engineering process began with a simple instruction, starting from the basic prompt "Résume :" to initiate the summarization. We then progressively enhanced the prompt by adding additional instructions to better guide the model toward producing the desired output.

We quickly discovered that a bare instruction was insufficient to obtain coherent and detailed summaries. To improve the quality, we modified the prompt to include more context and guidance. For example, one of our refined prompts was the one we used to generate our data with bigger models:

```
"Vous êtes un expert en synthèse de texte.  Veuillez fournir un résumé détaillé et complet
du texte suivant.
Assurez-vous d'inclure tous les détails importants et l'essence générale du contenu
en passant des lignes pour la lisibilité.
Texte :  "
```

This enhanced prompt provided clear instructions on what was expected, helping the model to focus on extracting and rephrasing essential content from the source text. We experimented with several variations—such as asking the model to explicitly "Ajoute les références au code civil si pertinent " or to emphasize certain points—to determine which formulation produced the most coherent and complete summaries.

Our analysis indicates that the prompt modifications led to a marked improvement in summarization performance for the mT5 model, as evidenced by significantly enhanced metrics when evaluated against reference summaries compared to generated summaries.



However, an additional challenge we encountered was the model's tendency to generate false references when explicitly instructed to quote official sources. When the prompt demanded direct citations—particularly for civil code references—the model sometimes produced plausible yet inaccurate quotations.

3.5 Fine Tuning

Our experiments reveal that a carefully fine-tuned small model can outperform a much larger model. As illustrated in Figure 2, strategic training and parameter optimization enable the smaller model to capture the nuances of the task effectively, demonstrating that model size alone does not guarantee superior performance.

Building on this observation, we find that when fine-tuned under identical parameters, both T5 and mT5 achieve nearly equivalent performance. This convergence suggests that T5 rapidly narrows its initial gap in handling English texts, matching the specialized capabilities of mT5, as shown in Figure 2.

Moreover, our results indicate that even though the models are trained on generated summaries, their similarity to the reference summaries—reflecting increased relevance and quality—improves steadily with additional epochs. This trend highlights the benefits of extended training in refining summary quality.

To further analyze the impact of training data on performance, we compare the fine-tuned T5 LoRA model when trained on LLM-generated summaries versus human-annotated reference summaries. As depicted in Figure 4, the model trained on reference summaries consistently achieves higher scores across all metrics, suggesting that while LLM-generated summaries can be useful, they may also introduce biases or errors that limit overall performance.

Then, we compare the performance of a fully fine-tuned T5 model with that of a LoRA-tuned T5 model. Figure 3 presents evaluation scores across key metrics such as ROUGE-1, ROUGE-2, ROUGE-L, and BLEU. Interestingly, full fine-tuning provides a slight edge over LoRA in most cases, particularly in capturing nuanced textual relationships as reflected by ROUGE-L. Nonetheless, the performance gap remains narrow, underscoring the efficiency of LoRA fine-tuning in achieving competitive results with significantly fewer trainable parameters.

Finally, the following figure summarizes our overall results, highlighting the performance of T5 and mT5 models under different training strategies. It clearly illustrates how, through strategic fine-tuning, both models converge toward high evaluation metrics, confirming that careful parameter optimization can enable smaller models to achieve competitive, and in some cases superior, performance compared to larger counterparts.

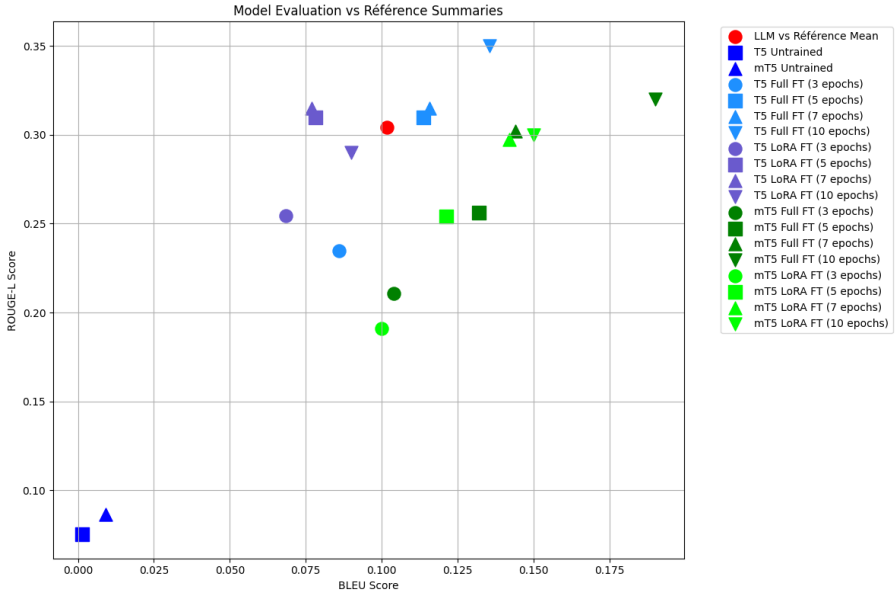


Figure 1: Overall Performance of T5 and mT5 Models Across Training Strategies.

4
FUTURE WORK AND PERSPECTIVES

Although our current methodology yields promising results in summarizing French legal texts, several avenues remain open for further exploration and enhancement.

For instance, advanced model architectures or hybrid approaches that combine the strengths of autoregressive and seq2seq models could be investigated to improve context integration and overall summary quality. Enhanced fine-tuning strategies, such as curriculum learning or reinforcement learning from human feedback, might also better capture the nuances of complex legal documents.

Additionally, integrating external domain-specific knowledge—using legal ontologies or knowledge graphs—could further enrich the model's understanding, leading to more comprehensive and contextually accurate summaries.

To address our challenge with hallucinations regarding civil code references, we could implement dynamic prompt engineering. By adapting the prompts based on the complexity and content of the input text, this approach offers a promising method to tailor outputs and mitigate the risk of generating erroneous or fabricated references. Recent works on automatic prompt engineering [4] and prompt tuning [2] provide strong evidence that optimized prompts can significantly enhance model performance, making them a valuable direction for future investigation.

Finally, it is important to recognize that training models with the objective of making them as good as human-crafted summaries can inadvertently lead to the learning of inherent biases if present un training data, which may have severe consequences when applied in critical domains such as law. This ethical consideration underscores the need for careful scrutiny of the fine-tuning process and the exploration of strategies that mitigate bias while maintaining model performance.

5
APPENDIX

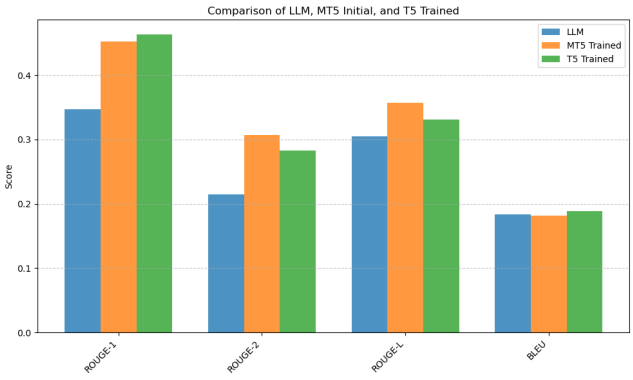


Figure 2: Performance of trained small models against an untrained bigger one.

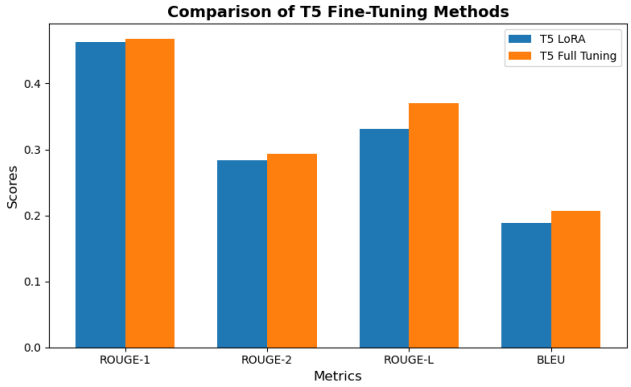
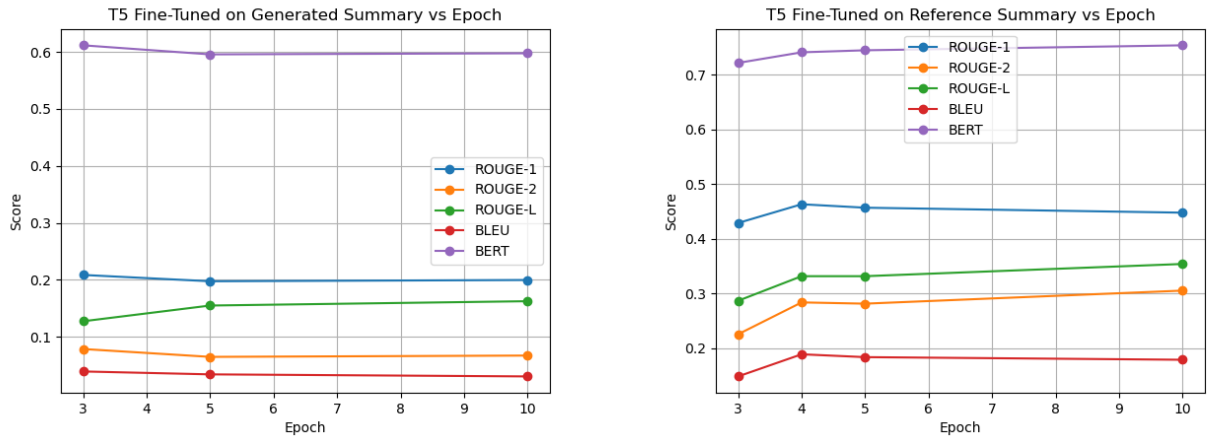


Figure 3: Performance of the fully trained T5 model vs LoRA-trained (on 10 epochs).



(a) Performance of the T5 model LoRA trained with LLM-generated summaries across different metrics.

(b) Performance of the T5 model LoRA trained with reference summaries across different metrics.

Figure 4: Performance of the T5 model fine-tuned with LoRA.

REFERENCES

[1] First Fu, John Doe, and Jane Smith. Comparing autoregressive and encoder-decoder architectures for conditional generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[2] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of EMNLP 2021*, 2021.

[3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[4] Taylor Shin et al. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of EMNLP 2020*, 2020.