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

CSC_52082_EP : Text Mining and NLP

14 mars 2025

Mindiiarova Renata & Boudelaa Fares

ÉCOLE POLYTECHNIQUE

INSTITUT POLYTECHNIQUE DE PARIS

TYPOGRAPHIX

TABLE DES MATIÈRES

1 INTRODUCTION

Text summarization is conceptually simple yet challenging to execute well. The objective is to retain as much important information as possible in the fewest words, requiring careful decisions about which information to exclude and which to emphasize. This task has broad applications across domains like summarizing patient ailments in medicine, or condensing court hearings in law. It is expensive when done manually, requiring high language expertise and usually domain-specific knowledge. With competent language models, this task becomes cost-effective without sacrifices in quality.

Today, language models are primarily trained using high-quality English data, making zero-shot summarization already quite effective for both small and large models in English. In other languages, especially low-resource ones, large language models might still perform relatively well without specific training due to their scale, despite not fully understanding linguistic subtleties and writing styles. Small language models, however, struggle significantly with summarization in non-English languages, thus requiring additional fine-tuning.

Fine-tuning a small language model for text summarization involves multiple steps : data selection and preparation, generation of high-quality summaries using larger language models, model selection and fine-tuning, and evaluation. We chose French, a language that isn't quite low-resource but due to its complicated grammar and verb structure, makes summarization less performant than one might expect.

Each decision we make has a specific justification : For data selection and preparation, we choose to use web-scraped Wikipedia articles, which are cleaned and shortened as needed. The generation of high-quality summaries is done using `'unsloth/mistral-7b-instruct-v0.3-bnb-4bit'`. Multiple models were chosen for fine-tuning using techniques such Low Rank Adaptation and quantization and we evaluated using ROUGE, BERTScore, BLEU, and LLM-as-a-judge metrics.

Although evaluation is highly subjective, our implementation shows promising results using a language model capable of inference on very low-resource devices, making it useful in real-world applications.

In the following sections, we detail our methodology for data collection and annotation, present our fine-tuning approach and implementation details, analyze our evaluation results against established baselines, and discuss the implications of our findings.

2 METHODOLOGY

2.1 Dataset Selection and Preparation

As mentioned in the introduction, the chosen language for this project is French. While French is not as low-resource as languages such as Greek or Japanese, its complex grammar and verb structures pose challenges for summarization tasks.

For dataset creation, we opted to construct our own dataset instead of relying on existing corpora. We developed a web crawler using Python, leveraging the `requests` library to fetch webpages and `BeautifulSoup` to parse and extract relevant text from HTML content. Our approach was systematic : we selected nine broad Wikipedia articles (‘Mathématiques‘, ‘Droit Civil‘, ‘Histoire de France‘, ‘Littérature Française‘, ‘Informatique‘, ‘Économie‘, ‘Finance‘, ‘Histoire‘, and ‘Philosophie‘) and then extracted all linked Wikipedia articles from these initial sources. This ensured a diverse dataset covering various general topics.

The collected raw articles required cleaning and preprocessing. For instance, many Wikipedia articles start with the phrase "Pour les articles homonymes, voir [...]," which serves as a disambiguation note. To remove such irrelevant sentences, we employed `spaCy` with the French pipeline `fr\_core\_news\_sm`. Additionally, we imposed a character limit of 4,000 (approximately two pages), truncating text at the nearest period. This decision stemmed from experiments with our chosen summarization model, which has constraints on input length. Given that summarization only considers the provided text, this truncation should not affect summary quality.

Our final dataset consists of 5,078 documents. Initially, the raw data had a mean length of 21,284 characters and a median word count of 3,365, with a maximum of 59,206 characters, far too long for small language models. After preprocessing, the dataset was significantly reduced, with a mean length of 3,261 characters and a median word count of 555, making it more practical for fine-tuning within computational constraints.

2.2 Synthetic Summary Generation

The synthetic summary generation uses a much larger model to generate high-quality summaries for supervised learning. These summaries serve as the golden standard for our fine-tuned model. For this task, we chose `'unsloth/mistral-7b-instruct-v0.3-bnb-4bit'`.

This choice followed experimentation with `'meta-llama/Llama-3.1-8B'`, `'Qwen/Qwen2.5-14B'`, and `'mistralai/Mistral-7B-Instruct-v0.3'`. The Meta model produced good results but occasionally answered in English, making it unreliable. The Qwen model offers excellent multilingual support but with costly inference. The Mistral model delivered the best time performance with high-quality summaries, making it optimal for this project. Furthermore, 4-bit quantization reduces memory requirements, enabling the model to fit on the GPU without issues. We also implemented vLLM, which leverages PagedAttention for dynamic KV Cache Management, increasing inference speed.

For prompt engineering, complicated queries weren’t necessary since the model was fine-tuned for instruction following. We structured the input as :

```
input = f"Texte :\n \n {text_to_summarize}\n \n RÃŀsumÃŀ en moins de 400 mots: \n "
```

We imposed a 400-word limit as a precaution, given our documents contain approximately 500 words on average. After running inference on our entire dataset, the average character length of our summaries is 478,

with an average word count of 155. This length is satisfactory for our task.

We also run a variant with a limit of 256 words, and use this prompt to generate even shorter summaries :

```
input = f"[INST] Texte :\n\n{text}\n\n REPONDS EN FRANCAIS. Resume ce texte en moins de 50
mots. Ne depasse pas la limite imposee et termine tes phrases correctement[/INST]\n"
```

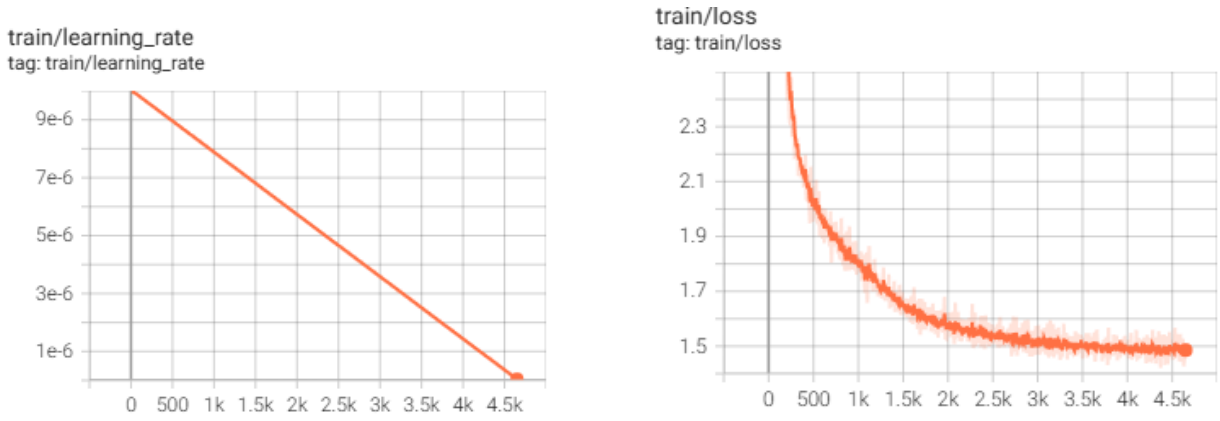
2.3 Model Selection and Fine-tuning

- **Selected SLM(s) and justification**

For the task of summarization, we selected the BARThez model, a French-specific adaptation of BART. Originally, BART (Bidirectional and Auto-Regressive Transformer) is a sequence-to-sequence (Seq2Seq) model designed for text generation and natural language understanding tasks. BARThez builds on this architecture by being pretrained using a denoising autoencoding approach on 66GB of French raw text, where it learns to reconstruct corrupted input sentences. Unlike BERT-based models such as CamemBERT and FlauBERT, which focus primarily on language understanding, BARThez is particularly well-suited for generative tasks like abstractive summarization, as both its encoder and decoder are pretrained to handle text generation more effectively.

- **Hyperparameter selection and optimization. Training details**

To optimize memory usage and speed up training, we quantized the BARThez model using BitsAndBytes-Config, enabling 8-bit precision. This reduced the model's memory footprint, making it feasible to fine-tune on standard GPUs without sacrificing performance.



(a) Training learning rate

(b) Training loss

Figure 1 – Training metrics

In addition to quantization, we applied Low-Rank Adaptation (LoRA) using the PEFT (Parameter-Efficient Fine-Tuning) framework with the following parameters : lora_alpha = 32, lora_dropout = 0.05, and lora_r = 16. The lora_alpha parameter controls the scaling factor for LoRA-adapted weights, ensuring a balance between adaptation and stability. The lora_dropout value of 0.05 helps prevent overfitting by introducing slight regularization during training. The rank parameter, lora_r, was set to 16, which determines the number of

trainable parameters in LoRA layers, allowing efficient adaptation while maintaining computational efficiency. This technique updates only a subset of model parameters, reducing computational costs while preserving fine-tuning effectiveness. We targeted key attention and feedforward layers, specifically "q_proj", "v_proj", "k_proj", "out_proj", "fc1", and "fc2", ensuring optimal adaptation for our summarization task.

For training, we set a learning rate of 1e-5 with weight decay of 0.01 to prevent overfitting. The model was fine-tuned for 150 epochs, with validation loss as the primary evaluation metric, as a lower loss indicates better performance. To manage GPU memory efficiently, we used gradient accumulation with a step size of 4, effectively increasing the batch size without exceeding hardware limits.

Training was conducted with a batch size of 32 per device for both training and evaluation. We enabled mixed-precision training (fp16) to further reduce memory consumption and improve computational efficiency. Logging was configured to occur every 10 steps for real-time progress tracking in TensorBoard.

2.3.1 • Evaluation Methodology

We used ROUGE to evaluate the quality of summarization. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measures n-gram overlap between the generated and reference summaries. Specifically, we used ROUGE-1, ROUGE-2, and ROUGE-L :

— **ROUGE-1** calculates the overlap of individual words.
— **ROUGE-2** considers bigram (two-word sequence) overlap.
— **ROUGE-L** evaluates the longest common subsequence.

These metrics help assess how closely the model-generated summaries match human-written ones, with higher scores indicating better performance.

In addition, we used **BERTScore**, which evaluates semantic similarity rather than exact word matching. Unlike traditional n-gram-based metrics, BERTScore leverages contextual embeddings from a pre-trained BERT model to compare the semantic closeness of the generated and reference texts. It computes three key values : **precision**, **recall**, and **F1-score**, where recall is particularly important for summarization tasks as it measures how much of the reference summary is captured in the generated output. BERTScore provides a more robust assessment by recognizing paraphrased content, making it a valuable complement to ROUGE.

In addition to traditional evaluation metrics such as ROUGE and BERTScore, we employed the **LLM-as-Judge** approach with Ollama to assess the quality of the generated summaries. The model evaluated each summary based on **coherence**, **factual accuracy**, and **overall quality**, assigning scores on a **1 to 5** scale. To ensure consistency in the evaluation, we used the following prompt :

```
"""
Vous êtes un expert en évaluation de résumés en français.
Votre tâche est de comparer un résumé généré à son texte original et d'évaluer sa qualité
selon les critères suivants :
Fidélité au texte original (1-5) : Le résumé transmet-il avec exactitude les informations
essentielles du texte original sans les déformer ni ajouter des éléments incorrects ?
Expliquez brièvement si certaines informations sont inexactes ou manquantes.
Exhaustivité (1-5) : Le résumé couvre-t-il toutes les idées principales du texte original ?
Justifiez si des éléments essentiels ont été omis ou s'il y a des répétitions inutiles.
Concision (1-5) : Le résumé parvient-il à exprimer les idées principales de manière concise
et efficace sans être trop long ni trop court ?
Indiquez si certaines parties sont trop détaillées ou, au contraire, insuffisamment développées.
Clarté et structure (1-5) : Le résumé est-il bien organisé et fluide, facilitant
la compréhension du texte ? Expliquez si l'ordre des phrases et la formulation rendent
le résumé plus ou moins lisible.
Score global (1-5) : Évaluez la qualité générale du résumé en tenant compte des critères précédents.
```

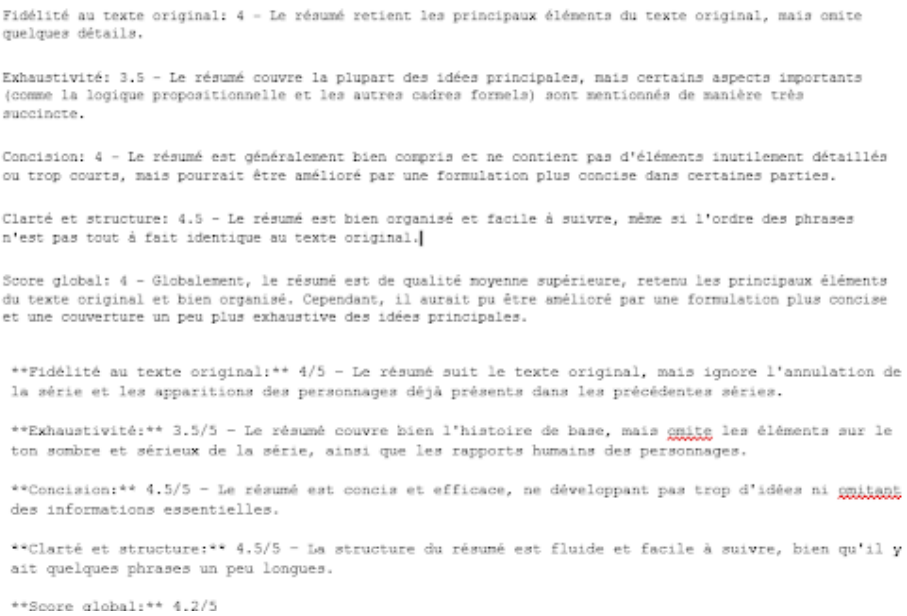


Figure 2 – LLM-as-a-Judge example

```
Texte original :
{test_data["text"].iloc[0]}
Résumé généré :
{generated_texts[0]}
Format de réponse STRICTEMENT attendu :
Fidélité au texte original: [Score] - [Courte justification]
Exhaustivité: [Score] - [Courte justification]
Concision: [Score] - [Courte justification]
Clarté et structure: [Score] - [Courte justification]
Score global: [Score] - [Courte justification]
"""
```

Due to the expensive computational requirement we only provide a single example of the ollama model evaluation :

3 RESULTS AND ANALYSIS

3.0.1 • Initial Overview of the Generated Data

The results of our training show promising outcomes. Starting with synthetic summaries, the Mistral-7B model generated high-quality summaries. The LLM doesn't hallucinate and maintains any logical or chronological order present in the full text. However, it sometimes failed to abide by the word limit constraint, occasionally stopping mid-sentence. As seen in our examples, the generation sometimes stops abruptly, such as after the phrase '. Dans'. This issue could be resolved by increasing the token limit, which would defeat the purpose of our project, so we left it as is. After experimenting with truncation and increasing the max token hyperparameter, we found no significant improvement in the fine-tuning process.

The fine-tuned BARThez model also outputs promising results. After reviewing random samples, we observed that the text maintains semantic meaning and preserves the ordering found in the original text. However, it also suffers from incomplete sentences due to token limitations.

Looking closely at the given example, where the original text can be found in the appendix, we can see that, compared to the reference, the generated text is more concise, uses fewer "bloat words," and prioritizes more direct phrasing. For example, instead of using "désigne," the model uses "est." This improves clarity and readability by making the text easier to understand while maintaining its original meaning.

Additionally, the model enhances precision by replacing vague or overly formal phrases with more straightforward alternatives. This ensures that the summarization remains effective without unnecessary complexity. Importantly, despite the reduction in word count, the core meaning is retained, demonstrating that the model successfully balances brevity with informativeness.

These remarks are based on qualitative observations ; let's now explore more quantitative results.

Metric	Rouge 1	Rouge 2	Rouge L	Bert Precision	Bert Recall	Bert F1
Score before	0.39	0.23	0.28	0.69	0.89	0.7
Score after	0.41	0.25	0.30	0.83	0.88	0.85

Table 1 – Evaluation Scores for Summarization using BARThez

Using the aforementioned metrics to compare BARThez output with the Mistral-generated summaries, we observe impressive BERTScore results. After fine-tuning, BARThez achieves a remarkable 85% on the BERTScore F1 measure, indicating strong semantic similarity between its outputs and the target summaries. This performance is particularly noteworthy considering BARThez is a significantly older and smaller model than Mistral, trained on substantially less data, which underscores the effectiveness of our fine-tuning approach.

Regarding ROUGE scores, which measure lexical overlap, the model achieves 41% on ROUGE-1, demonstrating considerable unigram overlap with reference summaries. For ROUGE-L and ROUGE-2, measuring longest common subsequence and bigram overlap respectively, BARThez attains scores of 30% and 25%. These results align with expectations given the architectural differences between the models and their distinct pre-training datasets. The moderate ROUGE-2 score suggests that while the model captures individual key terms well, it may occasionally structure phrases differently than the reference summaries.

Comparing our fine-tuned model to the original, we observe a significant increase in BERT F1 and BERT precision. This suggests that the base model was undertrained before fine-tuning. The notable boost in precision is likely due to the model generating more accurate and contextually appropriate wording. Additionally, fine-tuning helps reduce hallucinations, as seen in some examples. We also observe minor improvements in ROUGE scores, further indicating that the fine-tuned model is better suited for this summarization task and maintains fidelity to the reference text without drastically changing n-gram overlap patterns.

4 CONCLUSION

The fine-tuning of BARThez improved its summarization capabilities, achieving higher ROUGE and BERTScore metrics. The integration of Low-Rank Adaptation (LoRA) and quantization techniques effectively enhanced model performance, achieving results comparable to much larger language models while maintaining computational efficiency.

4.1 Key Findings

- **Efficient Fine-Tuning :** The use of LoRA and 8-bit quantization enabled full fine-tuning on standard GPUs, reducing memory requirements while maintaining model accuracy.
- **Improved Semantic Retention :** The evaluation metrics showed notable improvements, with increased ROUGE-1 and ROUGE-2 scores and a BERTScore F1 reaching 85%, indicating strong alignment with reference summaries.
- **Challenges in Output Length :** The model occasionally produced truncated summaries due to token limitations, highlighting the need for improved length control mechanisms.

4.2 Future Directions

This study demonstrates that fine-tuning smaller language models for summarization is feasible even with limited computational resources. Future work could explore :

- Further optimization of token constraints to prevent truncation issues.
- Domain-specific fine-tuning to enhance summarization quality in specialized fields.
- Improved evaluation strategies that combine both automated metrics and human-like assessments.

APPENDIX

	Count	Mean	Std	Max
Character Length	5078	21284.99	31760.62	366907
Word Length	5078	3365.52	5057.39	59206

Table 2 – Statistics for the original text.

	Count	Mean	Std	Max
Character Length	5078	3261.39	1116.90	4084
Word Length	5078	478.63	171.71	687
Summary Character Length	5078	1060.47	310.38	5248
Summary Word Length	5078	155.01	44.31	308

Table 3 – Summary statistics for the generated text.

	Count	Mean	Std	Max
Character Length	5078	1534.89	433.12	3708
Word Length	5078	221.71	54.37	256
Summary Character Length	5078	457.05	136.60	994
Summary Word Length	5078	70.11	20.93	147

Table 4 – Summary statistics for the second variant.

Other models

- ### Training

We also finetuned the new `"google/gemma-3-1b-pt"` released the 12th march 2025, as second model, using the shorter variant of the our dataset due to size of the model. We adapte the *tutoriel* published by google because the transformer libraire is not fully compatible with the model. A QLoRA fine-tuning approach was implemented to minimize memory requirements, using 4-bit quantization. Key hyperparameters include a learning rate of 2e-4, LoRA rank of 8, and alpha of 16, with training conducted over 3 epochs, with a maximum sequence length of 600 tokens because of the memory constraints.

- ### Quick analysis of the gemma model

Here, we briefly discuss the results of fine-tuning the Gemma model. Due to the length constraints of this report, we leave further analysis to the examiner as an additional model to evaluate. The Gemma model already exhibits strong out-of-the-box multilingual capabilities across more than 140 languages, producing high-quality initial summaries. However, like many smaller language models, it struggles with instruction adherence. When it does follow instructions, the summaries can be well-formed but sometimes lack coherence. A quick qualitative analysis of the data suggests that fine-tuning significantly improves summary quality, leveraging the model's extensive pretraining to generate more coherent and precise outputs. *The code and data are located here.*

RÉFÉRENCES

[1] Hu, Edward J., et al. “Lora : Low-Rank Adaptation of Large Language Models.” arXiv.Org, 16 Oct. 2021, arxiv.org/abs/2106.09685.

[2] Eddine, Moussa Kamal, et al. "BARThez : a Skilled Pretrained French Sequence-to-Sequence Model." arXiv.org, 23 Oct. 2020, arxiv.org/abs/2010.12321.

[3] Lewis, Mike, et al. "BART : Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7871-7880.

[4] Liu, Yinhan, et al. "Multilingual Denoising Pre-training for Neural Machine Translation." Transactions of the Association for Computational Linguistics, vol. 8, 2020, pp. 726-742.

[5] Jiang, Albert Q., et al. "Mistral 7B." arXiv.org, 17 Oct. 2023, arxiv.org/abs/2310.06825.

[6] Lin, Chin-Yew. "ROUGE : A Package for Automatic Evaluation of Summaries." Text Summarization Branches Out, 2004, pp. 74-81.

[7] Zhang, Tianyi, et al. "BERTScore : Evaluating Text Generation with BERT." International Conference on Learning Representations, 2020.

[8] Papineni, Kishore, et al. "BLEU : a Method for Automatic Evaluation of Machine Translation." Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp. 311-318.

[9] Dettmers, Tim, et al. "SpQR : A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression." arXiv.org, 5 June 2023, arxiv.org/abs/2306.03078.

[10] Kwon, Woosuk, et al. "Efficient Memory Management for Large Language Model Serving with PagedAttention." Proceedings of the 39th IEEE International Conference on Data Engineering, 2023.