Multilingual Summary Generation
Crosslingual comparison on finetuned Models

Shashwat SHARMA
Institut Polytechnique de Paris
shashwat.sharma@ip-paris.fr

Kazeto FUKASAWA
Institut Polytechnique de Paris
kazeto.fukasawa@ip-paris.fr

Abstract

In this study, We investigated different model architectures for multilingual summarization using a dataset constructed from Wikipedia 'good' and 'featured' articles in English, French, German, and Japanese. We compared three approaches: full-parameter quantised fine-tuning of Qwen2.5-0.5B-Instruct (494M parameters), qLoRA adaptation of Phi-4-mini-instruct (3.8B parameters), and traditional fine-tuning of mBART-50 (610M parameters) as an encoder-decoder baseline. Surprisingly, the fully fine-tuned Qwen model outperformed both alternatives, including the significantly larger Phi model, challenging conventional wisdom about parameter-efficient tuning methods. Our cross-lingual analysis revealed substantial knowledge transfer between languages, with models fine-tuned on a single language showing impressive improvements when generating summaries in other languages—particularly when summarizing semantically related topics in different languages, with ROUGE score improvements reaching 167.4% for French. These findings, validated through both ROUGE metrics and LLM-based evaluation with Gemma 3 27B, demonstrate that strategic fine-tuning of compact decoder-only models can provide effective multilingual summarization capabilities.

1 Introduction

1.1 Dataset Collection

The dataset consists of 'good' and 'featured' Wikipedia articles in four languages: English, French, German, and Japanese. We chose to use Wikipedia articles because we could get good transliteration of the same topic in different languages. The articles are well structured, so even when the article length is much longer than the model's input token limit, the initial paragraph contain overall information about the topic. Finally, the articles span over diverse domains from politics, science, music, etc.

We extracted the articles using 'mwclient' library in python. Each article was stored in a JSON format containing the article ID, title, full text content, and URL reference. As the articleID remains consistent for the same topic in different language, this would let us have much higher control when we evaluate to either prevent data leakage when comparing cross-lingually or monitor if the model captures semantic information from one language and could transliterate and produce better summaries in a different language.

The diversity of languages was crucial for this project, as it allows us to evaluate the cross-lingual capabilities of the fine-tuned model and ensures the robustness of our approach across different linguistic contexts.

NeurIPS 2023 Generative AI and Biology Workshop.

1.2 Summary Generation

1.2.1 LLM

To create high-quality reference summaries for training, We employed the Mistral Small 24B model, which represents state-of-the-art capability in the "small" LLM category (below 70B parameters). This model was chosen for several key reasons: 1. Exceptional multilingual capabilities, supporting all four target languages 2. Knowledge-dense architecture with strong reasoning capabilities 3. Large (32k) context window and 131k vocabulary size tokeniser

We used 4 bit quantization with double quantization, as shown in the Tab 1. We will use this quantization configuration again when we load the smaller language model.

1.2.2 Prompt Template

Prompting was particularly important, as without it, the generated document often contained unnecessary information, such as:

- Repeating the same summary twice.
- Including a summary in a different language, especially in English.
  - For example: “... Summary in English: Józef Piłsudski was the father of the Polish state...”
- Explaining the summary with phrases like “This summary is comprehensive and captures all key points.”
- Including instructions such as “If you need further assistance or a specific aspect to be highlighted, please let me know.”

By explicitly incorporating prompts to remove any extraneous information, we were able to improve the quality of the generated summary for labeling.

Initially, we used English prompts to generate summaries in different languages by instructing the model to produce summaries in French or German. However, this approach often resulted in summaries that still contained some English text. To improve accuracy, we switched to using prompts in the target language, which enhanced the quality of the generated teacher summaries. For each language, We designed tailored prompts that specifically instructed the model to generate concise but comprehensive summaries. An example prompt we used is shown in the appendix (Section 7).

To reduce potential biases and increase diversity in the generated summaries, We systematically altered the prompts after every 1,000 samples. This approach introduced slight variations in tone and structure while maintaining consistency in the overall quality and content of the summaries.

1.3 Data Preprocessing

The raw summaries generated by Mistral required minimal but important post-processing. Despite clear instructions to output only the summary content, simpler prompts occasionally resulted in the model appending notes or explanatory text. We implemented a cleaning step to remove these artifacts, ensuring that only the actual summary content was retained for training.

Quality control was implemented through random sampling and manual inspection of generated summaries across all four languages. This verification confirmed that summaries captured the essential information from the source articles, maintained language quality across all target languages, remained appropriately concise yet comprehensive, and contained no hallucinated content.

1.4 Dataset Organization

For optimal training efficiency, We converted the raw JSON data into the HuggingFace Dataset format using the Arrow data format. This choice improved memory efficiency through Arrow’s columnar memory layout, enabling memory-mapping for handling large datasets on machines with limited RAM. It also enhanced processing speed, providing 1–3 Gbit/s access speeds even on standard hardware. Additionally, batch processing was accelerated through HuggingFace’s batch mapping

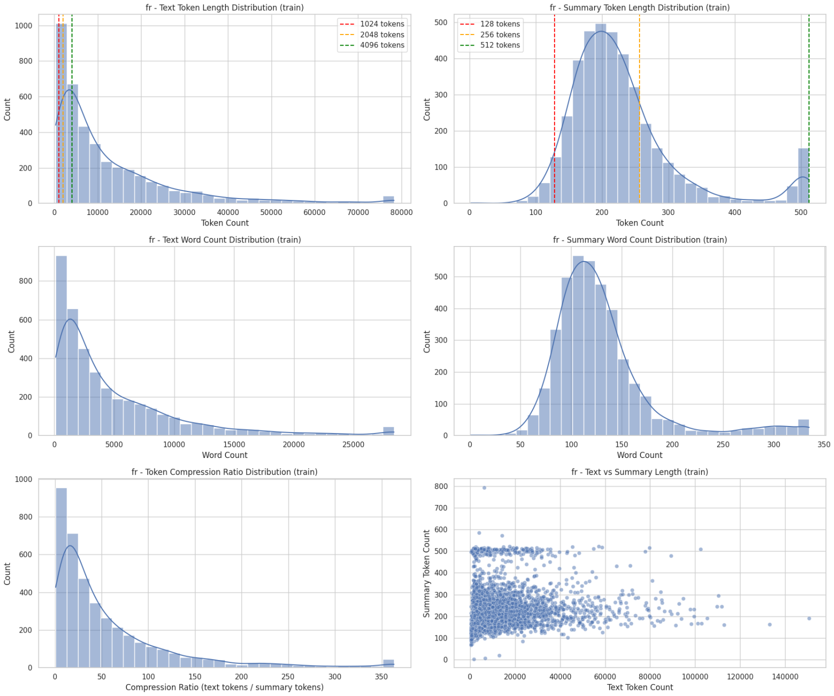

Figure 1: Text and Summary Token length statistics

capabilities, streamlining tokenization and preprocessing. The format's interoperability ensured seamless compatibility with PyTorch and other ML frameworks.

The dataset was structured with the following features:

- **article_id**: Unique identifier for each article
- **language**: The language code (en, fr, de, ja)
- **title**: The article title
- **text**: The full article content
- **summary**: The generated reference summary
- **url**: Source URL of the original article

Following best practices, We split the data into training (80%), validation (10%), and test (10%) sets, ensuring stratification by language to maintain equal representation across splits:

1.5 EDA

The input and output token lengths are shown in the Fig 6. The compression ratio represents the ratio of the number of tokens in the output summary to the number of tokens in the input. We can see a large number of summaries fall below 256 token limit so this is what we selected as output_token_length for the smaller language models. The wikipedia articles extracted have a lot of tokens because of html tags we talk a bit about this in the discussions section at the end.

2 Methodology

2.1 Experimental Design

Our project explores the trade-offs between full-parameter fine-tuning of smaller language models versus parameter-efficient fine-tuning of larger models for multilingual summarization. This comparison addresses a fundamental question in practical NLP deployment: is it more effective to fully fine-tune a compact model or to partially fine-tune a larger, more capable model? To investigate this question systematically, we designed a controlled experiment using models with distinct architectural profiles but trained on identical multilingual summarization data. We also explore whether an Encoder-Decoder architecture could outperform Decoder only architectures on the summarization task.

2.2 Model Selection

This selection allows us to evaluate whether the inherent capabilities of larger models can be effectively leveraged through parameter-efficient fine-tuning methods, compared to comprehensive training of all parameters in smaller models. We selected these transformer-based language models with different parameter scales and architectures:

2.2.1 Decoder Only Model

Qwen2.5-0.5B-Instruct: A compact multilingual model (494M parameters) designed for instruction-following tasks with relatively modest computational requirements. For more details, refer to the technical report: Qwen2.5 Technical Report (**?**).

Microsoft Phi-4-mini-instruct: A larger model (3.8B parameters) with enhanced reasoning capabilities and broader knowledge representation. Detailed information can be found in the technical report: Phi-4-Mini Technical Report: Compact yet Powerful Multimodal Language Models via Mixture-of-LoRAs (**?**).

2.2.2 Encoder-Decoder Model

For the encoder-decoder model, we selected mBART-50(2), as BART is a widely used encoder-decoder model, and mBART-50 is a multilingual version. It is a multilingual sequence-to-sequence model with 610 million parameters. mBART-50 is an extension of the original mBART model, incorporating 25 additional languages, enabling multilingual machine translation for 50 languages, including the languages we used in this project: English, French, German, and Japanese.

2.3 Finetuning

2.3.1 Full-Parameter Fine-tuning (Qwen)

In our investigation of full-parameter fine-tuning for the Qwen model, we adjusted all 494 million parameters during training. This comprehensive adaptation allows the model to fully align with the target task domain, potentially enhancing performance by enabling unrestricted parameter updates and removing limitations associated with frozen parameters. To manage memory usage effectively while updating all parameters, we employed gradient checkpointing, resulting in VRAM usage of approximately 14,076MiB during training. This approach aligns with findings from the study "Labeling supervised fine-tuning data with the scaling law" by Huanjun Kong, which demonstrated the viability of fine-tuning large language models for downstream natural language processing tasks.

2.3.2 Low-Rank Adaptation (Phi)

Low-Rank Adaptation (LoRA) was employed for the larger Phi-4 model as a parameter-efficient fine-tuning technique. As introduced by Hu et al. (2021), LoRA strategically inserts trainable low-rank decomposition matrices into specific layers while freezing the majority of parameters. This approach offers significant advantages: drastically reduced memory requirements, accelerated training with fewer trainable parameters, and the ability to preserve the model's pre-trained knowledge while adapting to new tasks. For our implementation, we utilized LoRA to minimize VRAM usage with carefully selected hyperparameters as detailed in next section. This method, further extended by Dettmers et al. (2023) with QLoRA, enables efficient fine-tuning of large pre-trained models on new tasks without modifying or storing substantial portions of the model's weights, making it particularly valuable for resource-constrained environments.

2.3.3 Encoder-Decoder Fine-tuning (mBART-50)

We fine-tuned mBART-50 on Wikipedia data without using LoRA or quantization.The architectural differences of encoder-decoder models, which may benefit differently from full parameter updates. The fine-tuning process required 15.8 GB of VRAM, which is within the free-tier resources available on Google Colab. The model was trained for three epochs, with each language taking approximately 20 minutes for three epoch finetuning. Here, we used MeCAB for word segmentation in Japanese before computing the ROUGE score.

2.4 Training Configuration

2.4.1 Common Training Parameters

shared several training parameters to ensure fair comparison:

- **Input Sequence Length**: 4096 tokens
- **Summary Length**: 256 tokens
- **Optimizer**: AdamW with fused implementation
- **Learning Rate Scheduler**: Cosine with warmup
- **Batch Accumulation**: Gradient accumulation to achieve effective batch sizes
- **Evaluation Strategy**: Per-epoch evaluation with ROUGE metrics

2.4.2 Model-Specific Training Parameters

Qwen2.5-0.5B:

- **Learning Rate**: 2e-5 (lower to prevent divergence with full parameter updates)
- **Batch Size**: 1 with 16 gradient accumulation steps
- **Weight Decay**: 0.01
- **Precision**: Mixed precision BF16

Phi-4-mini:

- **Learning Rate**: 5e-4 (higher learning rate feasible with LoRA)
- **Batch Size**: 1 with 16 gradient accumulation steps
- **Weight Decay**: 0.01
- **Precision**: BF16 with 4-bit quantization
- **LoRA Rank (r)**: 32
- **LoRA Alpha**: 64
- **LoRA Dropout**: 0.05

mBART-50:

- **Learning Rate**: 3e-5
- **Epochs**: 3 per language
- **Precision**: FP32 (no quantization applied)

2.5 Prompt Template

We maintained consistent prompting templates across both models to ensure valid comparisons. The prompts were carefully designed to be language-specific, with explicit instructions to generate concise yet comprehensive summaries. For example, the English prompt template is shown below. Similar templates were crafted for French, German, Japanese, and Russian, using native language instructions to enhance performance.

```
Please provide a concise summary of the following article in English.
The summary should be comprehensive, capturing all key points and main arguments,
but avoid unnecessary details. Output only the summary.

Article:
{text}

Summary:
```

2.6 Evaluation Metrics

During training, we implemented an automated evaluation system using ROUGE metrics (ROUGE-1, ROUGE-2, ROUGE-L) with a custom RougeEvaluationCallback. This callback randomly sampled validation examples, generated summaries using the current model state, and compared them against reference summaries. The results were logged to Weights & Biases, providing continuous tracking of model performance and enabling the identification of convergence patterns or potential divergence issues.

We used ROUGE (Recall-Oriented Understudy for Gisting Evaluation)(1) score. ROUGE score is a set of metrics used to evaluate the quality of summaries by comparing them to reference summaries. It is widely used for tasks like automatic text summarization, machine translation, and text generation. ROUGE measures the overlap between the n-grams (unigrams, bigrams, trigrams, etc.) in the generated text and the reference text. Higher overlap indicates better performance. Several different types of ROUGE scores exist. For our evaluation, we used ROUGE-1 (unigram overlap), ROUGE-2 (bigram overlap), and ROUGE-L, which captures the longest common subsequence to assess fluency and coherence. In addition to ROUGE, we also employed LLM-based evaluation on the test samples to gain deeper insights into summary quality. The details of this evaluation approach are explained in the next section of the report.

The curious case for Japanese The ROUGE score for Japanese was unusually low because ROUGE is based on n-grams, which require separated text as input. However, Japanese text does not naturally include word separation. As a result, entire sentences were treated as single words, leading to an abnormally low ROUGE score for Japanese. To accurately evaluate performance in Japanese, a word segmentation algorithm would be necessary. So, we chose to use a custom Tokenizer to correctly evaluate the metrics on Japanese text. This is done using the MeCabTokenizer

2.7 Monitoring and Logging

We implemented comprehensive training monitoring through Weights & Biases integration, tracking parameters for transparency and reproducibility. Loss curves were used to monitor convergence patterns, while the ROUGE metric evolution was logged over training epochs. Example outputs were recorded alongside their corresponding metrics, and resource utilization statistics were monitored throughout the process.

3 Experimental Results

3.1 Model Comparison

To determine the most effective architecture for multilingual summarization, we conducted a comparative analysis between decoder-only models (Qwen2.5-0.5B and Phi-4-mini-instruct) and encoder-decoder architecture (mBART-50). Our evaluation revealed that while mBART-50 was successfully fine-tuned for the summarization task across all target languages, its performance plateaued at levels comparable to the pre-fine-tuned decoder-only models. Specifically, mBART-50's ROUGE scores after complete fine-tuning were similar to the baseline scores of Phi-4-mini before any adaptation, indicating a lower performance ceiling for the encoder-decoder architecture in this context.

The decoder-only models demonstrated superior performance trajectories, with substantial improvements after fine-tuning. The fully fine-tuned Qwen2.5-0.5B model and LoRA-adapted Phi-4-mini both achieved significantly higher ROUGE scores across all language pairs compared to the fine-tuned mBART-50. This performance gap persisted despite mBART-50's comparable parameter count to Qwen2.5-0.5B (610M vs. 494M parameters) and its explicit design for multilingual tasks. The superior performance of relatively compact decoder-only models suggests that recent advances in decoder-only architectures may have surpassed traditional encoder-decoder designs for certain multilingual text generation tasks, even with comparable parameter counts.

3.2 Cross Lingual Performance

To investigate the multilingual capabilities and knowledge transfer between languages, we conducted cross-lingual analysis using the Qwen2.5-0.5B model. This analysis explores how fine-tuning on

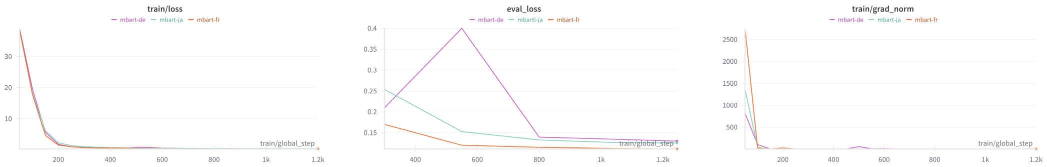

Figure 2: Training and Validation Metrics mBART

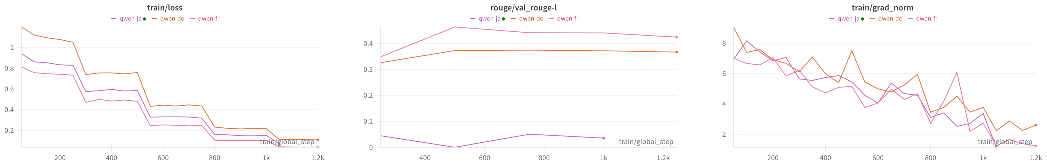

Figure 3: Training and Validation Metrics qwen

one language affects performance on other languages, so we can know about model's ability to transfer linguistic and structural knowledge across language boundaries. We fine-tuned separate Qwen2.5-0.5B models on summarization data in three distinct languages i.e. German (DE), Japanese (JA) and French (FR). Each language-specific model was then evaluated on generating summaries in three target languages i.e. German, English and French. The Performance is measured using percentage improvement in ROUGE-1, ROUGE-2, and ROUGE-L scores compared to the base model without fine-tuning. This evaluation was conducted on two distinct test sets: Novel Articles - completely unseen articles with different topics from the training data (test set). Same-Topic Articles - articles covering the same topics as those in training, but in different languages

As expected, models performed best when generating summaries in the same language they were fine-tuned on. The French-trained model showed the most dramatic improvements, with an 80.9% increase in ROUGE-1 scores on novel topics. Similarly, the German-trained model demonstrated strong gains, achieving a 60.7% improvement. The larger gains observed suggest that the model effectively learn language-specific patterns, which enhances their ability to generate accurate and coherent summaries.

The cross-lingual transfer effects reveal several intriguing patterns. Notably, knowledge transfer between language pairs is asymmetric—for instance, the German-trained model improved French summarization by 31.4% in ROUGE-1 on novel topics, whereas the French-trained model only boosted German summarization by 24.2%. The Japanese-trained model, on the other hand, exhibited relatively consistent improvements across all target languages, with ROUGE-1 gains ranging from 24.5% to 28.2%. This suggests that fine-tuning on Japanese, a language with a distinct grammatical structure and writing system, may have encouraged the model to develop more generalizable summarization capabilities rather than relying solely on language-specific patterns. Interestingly, all three fine-tuned models improved English summarization performance despite not being explicitly trained on English summaries, demonstrating cross-lingual transfer of summarization skills with ROUGE-1 improvements ranging from 22.9% to 26.7% on novel topics.

The comparison between novel-topic and same-topic results provides valuable insight into what the models are learning. The significantly higher improvements on same-topic articles, such as the 167.4% vs. 80.9% ROUGE-1 increase for French, suggest that the models are not only learning summarization structure but also capturing semantic content knowledge that transfers across languages. At the same time, the consistent improvements on novel topics indicate that the models are developing general summarization skills, including the ability to identify salient information, recognize structural patterns in Wikipedia articles across languages, and filter out unnecessary details such as metadata and tags. Additionally, they seem to avoid common summarization artifacts like self-referential text (e.g., "This summary covers..."). Notably, the Japanese-trained model demonstrated similar performance gains on both novel and same-topic articles, suggesting that it has learned more language-independent summarization features, possibly due to the distinct grammatical and structural properties of Japanese.

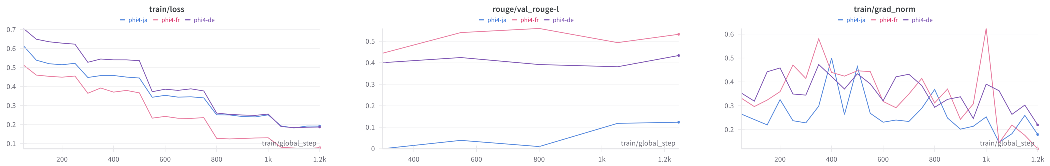

Figure 4: Training and Validation Metrics phi4

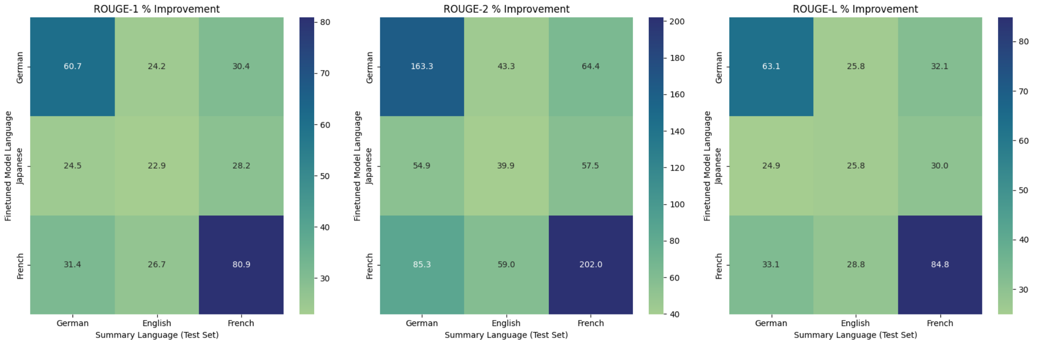

Figure 5: Performance of the ‘Novel Articles’

3.3 LLM as an evaluation

While ROUGE metrics provide valuable quantitative assessment of summary quality, they have notable limitations for evaluating multilingual summarization. ROUGE primarily measures lexical overlap through n-gram matching, which fails to capture semantic equivalence, factual accuracy, or linguistic quality.It penalizes semantically equivalent but lexically divergent expressions, which is especially problematic across languages. Additionally, language-specific biases arise due to differences in morphological structure, leading to systematically different scores that do not always reflect true summary quality.

So we implemented LLM as an evaluator for our summaries. The goal for the LLM is to take the base summary and the finetuned model summary in the input prompt and with structured output give us an Integer rating for both the summaries. 0 being the worst summary and 5 being the best summary possible. For this task we loaded the newly released ‘google/gemma-3-27b-it’ model which ranks highest for reasoning tasks on multiple benchmarks in its model parameter range. Additionaly, this model being different from ‘mistralai/Mistral-Small-24B-Instruct’ which we used to generate our base summaries from potentially reduces bias in evaluation as the same model could probably rate a summary higher which mimics the training tone of the original summariser.

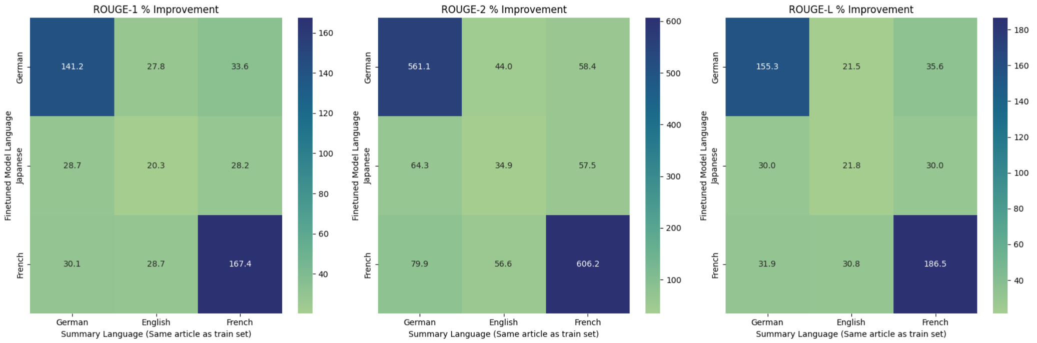

Figure 6: Performance of the ‘Same Topic Articles’

The fine-tuned models demonstrated strong intra-language improvements, with the German-trained model outperforming the base model by 1.75 points on test data and 2.0 points on training topics. The French-trained model showed even greater gains, improving by 1.65 points on test data and 2.45 points on training topics, reinforcing our earlier ROUGE findings that highlighted particularly strong adaptation to French. In terms of cross-lingual transfer, the German-trained model improved English summaries by 0.55–0.75 points and French summaries by 0.65–1.05 points, confirming its ability to generalize beyond its training language. Similarly, the French-trained model enhanced German summaries by 1.05 points and English summaries by 0.85–0.9 points, demonstrating robust transfer effects. Interestingly, the base model's performance varied by language, with English summaries receiving higher scores (around 2.7), suggesting stronger pre-fine-tuning capabilities in English. In contrast, German and French summaries scored lower (1.45–2.05), indicating greater potential for improvement through fine-tuning.

The LLM-based evaluation reinforced our ROUGE findings while offering additional insights. Notably, it highlighted that the base model produced significantly higher-quality English summaries pre-fine-tuning, a nuance not captured by ROUGE. Unlike ROUGE's relative improvements, LLM scores provided an absolute quality assessment, showing that fine-tuned models consistently generated strong summaries across languages (3.0–4.35). Additionally, cross-lingual gains remained substantial, with improvements of 0.55–1.05 points on a 5-point scale, emphasizing the effectiveness of multilingual fine-tuning even for languages not explicitly included in training.

4 Conclusion

In this study, we developed a high-quality multilingual dataset spanning four languages and evaluated different model architectures for multilingual summarization. After comparing a decoder-only model with an encoder-decoder model, we selected the decoder-only approach, due to its strong performance and efficiency. Our analysis showed that mBART, after fine-tuning, performed similarly to Phi-4-mini before fine-tuning. Further, full fine-tuning on the smaller Qwen-0.5B model significantly outperformed LoRA-based adaptation on the much larger Phi-4-3.84B model, achieving comparable final evaluation metrics with lower memory requirements, making Qwen the preferred choice for continued analysis.

Our findings highlight key implications for cross-lingual transfer. Fine-tuning a small model like Qwen-0.5B on a single language provides benefits across multiple languages, demonstrating efficient transfer learning. The model effectively captures both language-specific patterns and broader summarization skills, with the strongest transfer occurring when topics align with those seen during training, even in different languages. The varied effectiveness of transfer between language pairs suggests that linguistic similarity influences cross-lingual knowledge transfer.

The LLM-based evaluation further supports these conclusions, showing that strategically fine-tuning smaller models on select languages can yield substantial multilingual benefits. The alignment between ROUGE and LLM evaluations adds credibility to our findings, particularly in demonstrating strong cross-lingual performance on training topics. This suggests that for multilingual applications with overlapping content across languages, fine-tuning a smaller model on a single language may be an efficient and scalable approach, reducing the need for separate models while maintaining high summarization quality.

5 Discussion

The input Wikipedia article contains tags, as shown in the example below.

> In April 1986, the first [[savannah cat]], a hybrid between a male serval and a female [[domestic cat]], was born; it was larger than a typical domestic kitten and resembled its father in its coat pattern. It appeared to have inherited a few domestic cat traits, such as tameness, from its mother. This [[cat breed]] may have a dog-like habit of following its owner about, is adept at jumping and leaping, and can be a good swimmer. Over the years it has gained popularity as a pet.<ref>cite journal |author=Wood, S. |year=1986 |title=Blast from the Past: The Very First F1 Savannah |journal=Lioc-Escf |volume=30 |issue=6 |page=15</ref>

Although the LLM can ignore unnecessary tags when processing the article, it is preferable to train it on natural language only in this case. Additionally, unnecessary tags increase the length of the input tokens. However, as long as the LLM can handle long sentences that contain all necessary information from the article, it can still successfully summarize the input with HTML tags.

At the same time, some input tags may contain important information for summarization, and research suggests that using computer languages like JSON format can improve model performance. Therefore, comparing input data with and without HTML tags remains an area for future research.

References

[1] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[2] Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. Multilingual translation with extensible multilingual pretraining and finetuning. 2020.

6 Appendix: Tables

Parameter	**Value**
`load_in_4bit`	True
`bnb_4bit_use_double_quant`	True
`bnb_4bit_quant_type`	nf4
`bnb_4bit_compute_dtype`	torch.bfloat16

Table 1: Parameter settings for the BitsAndBytesConfig

Language	Model Type	ROUGE-1	ROUGE-2	ROUGE-L
French	Base Model	0.1235	0.0238	0.1101
	Fine-tuned Model	0.4351	0.2408	0.4153
German	Base Model	0.1605	0.0372	0.1451
	Fine-tuned Model	0.3104	0.1435	0.3083
Japanese	Base Model	0.1936	0.0699	0.1602
	Fine-tuned Model	0.4310	0.2375	0.3352

Table 2: mBART-50: ROUGE Scores for French and German Models

7 Appendix A: Prompting

English

```
Please provide a concise summary of the following article in English.
The summary should be comprehensive, capturing all key points and main
arguments,
but avoid unnecessary details. Output only the summary.


Article:
{text}


Summary:
```

French

```
Veuillez fournir un résumé concis de l’article suivant en français.
Le résumé doit être complet, capturant tous les points clés et les
arguments principaux,
mais évitant les détails inutiles. Ne produisez que le résumé.


Article:
{text}


Résumé:
```

Germany

```
Bitte erstellen Sie eine prägnante Zusammenfassung des folgenden Artikels
auf Deutsch.
```

```
Die Zusammenfassung sollte umfassend sein, alle Hauptpunkte und
Hauptargumente erfassen,
aber unnötige Details vermeiden.  Geben Sie nur die Zusammenfassung aus.


Artikel:
{text}


Zusammenfassung:
```

Japanese

```
以下の記事を日本語で簡潔に要約してください。
要約は包括的であり、すべての重要なポイントと主な議論を捉える必要がありますが、
不必要な詳細は避けてください。要約のみを出力してください。


記事:
{text}


要約:
```

8 Appendix B

```
target_modules = [
    "Wqkv", "out_proj",  # Phi-4 attention modules
    "up_proj", "down_proj"  # Phi-4 MLP modules
]

peft_config = LoraConfig(
    task_type=TaskType.CAUSAL_LM,
    inference_mode=False,
    r=32,
    lora_alpha=64,
    lora_dropout=0.05,
    target_modules=target_modules,
    bias="none",
)
```