March, 2025

CSC_52082_EP

Text mining introduction and NLP

Final Project Report

Name: Enzo Pinchon, Baptiste Geisenberger

Fine-tuning a Small Language Model for Text Summarization

Abstract

This report explores the fine-tuning of a Small Language Model (SLM) for Automatic Text Summarization (ATS), focusing on balancing efficiency and summary quality. To achieve this, we start by curating a dataset of 5,000 French texts from diverse sources, apply preprocessing steps, and carefully split it into training, validation, and test sets. Target summaries are generated using a pre-trained Large Language Model (LLM) and used to fine-tune two SLMs with three techniques (LORA, Prefix-Tuning and LORA with curriculum learning), optimizing hyperparameters through a dedicated tuning phase. Performance is evaluated using ROUGE, BERTScore, a LLM-as-judge approach, and a varation of RUSE score, benchmarking against the same SML before fine-tuning. The study highlights the effectiveness of fine-tuning SMLs for specific tasks, while also addressing practical challenges encountered during implementation and potential avenues for improvement.

Contents

1 Problem statement

In the past decades, we have observed an explosion of textual information available to individuals, especially accelerated by the rise of Internet. In this new era, individuals and organizations struggle to process this vast amount of texts efficiently, making automatic summarization essential in the majority of domains.

Automatic Text Summarization (ATS) aims to condense lengthy texts while preserving key information. The advent of LLMs has revolutionized this research domain by significantly enhancing linguistic comprehension, making them the most effective approach for generating high-quality, coherent summaries [8]. However, practical challenges remain. Most LLMs are optimized for English, leading to performance disparities in other languages [4, 14, 16]. Additionally, general-purpose LLMs are computationally expensive and difficult to deploy efficiently [17]. To address these limitations, researchers have explored a variety of techniques such as Cross-Lingual Transfer Learning to mitigate language disparities [10, 3], or Knowledge Distillation [6, 13], Prompt Engineering, and Fine-Tuning [8] to enhance the performance of lighter models. Our work focuses on the latter, exploring fine-tuning strategies on a Small Language Model (SML, defined as having 7 billion parameters or fewer) to develop an efficient ATS system for French texts. As requested and except stated otherwise, we run all our code on NVIDIA RTX A4000 GPU (16GB of VRAM). Across the projects, we often refer to quantization to fit in these constraints, which we implemented with the BitsAndBytes library from Hugging Face [1]. In the following sections, we present our methodology, detailing dataset curation, the fine-tuning of the SML, an extensive performance evaluation phase, and finally, a discussion on further directions for improving our method.

2 Curating a dataset

2.1 Data collection

A qualitative dataset along with a well-defined gold standard, is crucial for ensuring efficient training and effective summarization performance. It should be diverse enough to prevent overfitting, well-structured with consistent formatting, and free of redundancies such as pre-existing summaries that could bias the model's learning process. We refined our selection and filtering pipeline through automated and manual validation to ensure high relevance and quality.

2.1.1 Scraping a dataset from Wikipedia

Initially, we devised a Wikipedia scraping tool for geopolitical-related articles, thinking that limiting the domain improves the model's grasp of relevant sentence structures and concepts. To effectively narrow the domain, we started at the "Relations Interna-

[1] Hugging Face's BitsAndBytes quantization library

tionales" page and recursively scraped Wikipedia's subcategories, employing Mistral 7B [2] as a LLM-based filter, to assess whether a page was directly related to geopolitics. Additionally, we filtered out excessively long pages to prevent computational overhead. While this method showed promise, we encountered two key issues: (1) most scraped pages remained too lengthy, requiring a large context window for the SML, sharply increasing computational costs, and (2) many pages included a built-in summary at the beginning, making the summarization task less meaningful.

2.1.2 Lapresse dataset

To address these limitations, we opted for a second dataset sourced from [18], comprising 73,447 articles from Canadian newspaper LaPresse [3]. After a careful examination of the dataset, we selected the most relevant articles for our task. First, we identified multiple versions of the same articles and retained only the most recent ones to avoid redudancy. Next, we applied a length filter by computing token counts after tokenization with our SML, refining the dataset to a final selection of 5,073 articles. This dataset presented several advantages: it maintained thematic diversity while exhibiting a relatively uniform writing style, and crucially, it did not contain pre-existing summaries.

2.1.3 Data preprocessing

To ensure high-quality text for summarization, we performed a series of preprocessing steps following best practices [8]. First, we removed all HTML tags and URLs. Next, we stripped extra spaces and tabs, and we normalized line breaks. We also removed unknown symbols while preserving punctuation, ensuring the text retained its readability. These preprocessing steps helped standardize the dataset, reduced noise, and improved the quality of the input for fine-tuning our SML.

2.2 Summary generation

With the dataset curated, our next step was to define the gold standard. In order to achieve this, we evaluated two different large language models, namely **Qwen2.5-32B Instruct** and **Mistral 24B Instruct 2501**. These models were selected based on their contrasting architectural characteristics and performance profiles, presented in 1.

Table 1: Comparison of Qwen2.5-32B and Mistral 24B

Model	**Size (4-bit)**	**Exec. Time**	**Context Lengths**	**Languages**
Qwen2.5-32B	∼22GB	∼40s	128K	Multilingual (29+ languages)
Mistral 24B	∼15GB	∼15s	32K	Primarily English, multilingual

[2]Mistral 7B
[3]Canadian Newspaper Lapresse

Although Qwen2.5-32B was initially proposed in the final project instructions, it exceeds the 16GB VRAM constraint, even after after 4-bit double quantization. Nevertheless, we decided to include this model by running it on a NVIDIA RTX A5000 (24GB of VRAM), willing to provide a comparison of the two models. In contrast, Mistral 24B meets the size requirement and offers a significantly faster execution time. We then process the texts for summarization, contextualized with a carefully designed prompt manually tested on both models, ensuring a fair comparison between the two.

> **System Message:** *"Tu es un agent qui résume des textes en Français. Ta réponse doit être seulement le résumé du texte demandé. Contextualise le résumé si possible."*
>
> **User Input:** *"{Text to summarize}*
> *Résumé:"*

Since both models produced summaries with similar accuracy, we chose Mistral 24B for dataset generation due to its lower computational cost. The only exception was for paired curriculum learning 3.3, where we used summaries from both models to create the pairs.

2.3 Dataset split

We decided to split the dataset in three parts: train set, test set and validation set. To ensure a nice split and a representative dataset, we made sure to retain the same distribution in length of the initial texts for the each dataset split 2.

3 Fine-Tuning the Small Language Models

Having generated high-quality text-summary pairs, we developed a comprehensive fine-tuning strategy. This involves selecting appropriate models, implementing three fine-tuning techniques, for each of which we conduct a thorough hyperparameter tuning phase.

3.1 Choice of the Small Language Models

Our definition of a SML was one with fewer than 7 billion parameters, and we consulted [20, 21, 5] to carefully select our model. Many SMLs seemed promising, but presented problems. Ada Instruct v1 (350M parameters) [4] is unavailable in Hugging Face's Transformers library, Curie Instruct v1 (6.7B parameters) [5] is only accessible in a pre-fine-tuned version, and BART-Large (406M parameters) [6] has a context window that is too limited for our needs. Initially, we explored workarounds for BART-Large, drawn by its strong performance in [5]. We attempted a hierarchical summarization approach, recursively summarizing smaller text chunks before generating a final summary, but it significantly

[4] Ada Instruct v1
[5] Curie Instruct v1
[6] BART-Large

increased computation time and was difficult to implement with existing fine-tuning libraries. These challenges highlight the need for robust libraries and open models to enhance collaboration within research.

Prioritizing models with larger context windows, we selected two sizes: one around 3-4 billion parameters and another close to 7 billion. For the 3-4 billion category, we compared Phi-3.5-mini-instruct (3.8B) [7] and LLaMA-3.2-3B [8], as research shows outstanding performance of these two models for their size. We opted for LLaMA-3.2-3B, as from our research, it shows a slight edge [20, 15]. To fit within 16GB VRAM constraints, we quantized it to 8-bit using a computation threshold of 6.0, following Hugging Face recommendations. For the 7 billion parameter model, we chose Mistral-7B, known for its strong performance [1, 15], and quantized it to 4-bit.

3.2 Fine-tuning techniques

Given our GPU constraints, full fine-tuning did not seem relevant as too computationally heavy. We therefore decided to implement parameter-efficient tuning techniques: LoRA (Low-Rank Adaptation) [7, 19] and Prefix Tuning [11, 23]. LoRA freezes the original trainable parameters and updates them by attaching low-rank matrices to these weights, making it highly memory-efficient while still allowing deep task-specific adaptation. Following best practices, we applied this technique to the projection layers of the transformer block. Prefix Tuning is a lightweight and modular alternative [9], in which a small set of continuous task-specific prefix vectors prepended to the model's input representations is optimized. Less suited for highly task-specific adaptation like ours, this technique still provides valuable insights, approximating an optimal prompt and enabling an evaluation of the model's inherent capabilities without manual input design. We used a standard cross-entropy loss for training and attempted early stopping on the validation set, but GPU constraints prevented its implementation. We trained the Llama-3.2 3B for 5 epochs across both techniques, and the Mistral 7B for 3 epochs B.

For implementation, we used Hugging Face's Transformers library [9] to load and train the models and the PEFT library [10] to integrate the fine-tuning techniques. Running these models on 16 GPU required many practical parameter tweaks to decrease GPU usage. Standardizing the prompt aids in reducing variability in training, improving generalization across different samples. Therefore, we applied a standardized chat template:

```
<titre>:
<texte>:
<résumé>:
```

It was a shame to later discover that Hugging Face's PEFT library does not support combining LoRA and Prefix in one fine-tuning phase, which we had intended to explore.

[7] Phi-3.5-mini-instruct
[8] LLaMA-3.2-3B
[9] Hugging Face's Transformers library
[10] Hugging Face's PEFT library

The Adapters library [11] offers this functionality, implementing it is left for future work.

3.3 Curriculum learning with pairwise loss and LoRA

This method integrates curriculum learning with pairwise loss, guiding the model from simpler to more complex examples. ROUGE scores indicate summarization difficulty: higher scores suggest easier summarization, as key information is explicitly present, whereas lower scores (when accompanied by high RUSE or LLM-based scores) imply more challenging cases that require abstraction and rephrasing.

To implement curriculum learning, we sort the dataset by ascending ROUGE score and reduction factor, ensuring a gradual learning progression.

$$D = \frac{\text{LLM Score} + \text{RUSE Score}}{2(\text{ROUGE}_L + \epsilon)} + \lambda \cdot \text{Reduction Factor}$$

3.4 Hyperparameter tuning

For each model - fine-tuning technique combination, we conducted an extensive hyperparameter tuning phase, crucial for optimizing performance. Selecting appropriate hyperparameters significantly impacts training stability, convergence speed, and final model quality. To achieve this, we employed Bayesian optimization, a probabilistic approach that models the objective function and strategically selects hyperparameter values to maximize performance [2]. We implemented this process using the Optuna framework [12], which provides an adaptive and efficient way to optimize hyperparameters while integrating seamlessly with our fine-tuning pipeline.

4 Evaluation

After fine-tuning the models, we conduct an extensive evaluation phase to assess their performance and analyze both real and LLM-generated synthetic datasets for deeper insights.

4.1 Evaluation metrics

Rigorous evaluation is essential to assess model performance and gain insights into its summarization capabilities. However, evaluating summarization quality is a complex task, as it involves quantifying textual performance that are often subjective, such as

[11] Adapters library

[12] Optuna - a framework for bayesian hyper-parameter optimization

coherence, relevance, accuracy, non-redudancy and readability [17]. Therefore, we design the evaluation process using multiple metrics to comprehensively assess different aspects.

Our first two metrics are foundational in judging ATS: ROUGE [12], which captures lexical overlap but lacks semantic depth, and BERTScore [22], which computes token-level cosine similarities using pre-trained BERT embeddings. It uses greedy matching to find the most similar tokens, then averages these scores to calculate precision, recall, and F1. We also devise a LLM as judge approach, feeding each text-summary pair in Mistral 7B, through a carefully designed prompts asking to take into account the following factors: faithfulness, conciseness, clarity and context C.

Finally, we defined a custom-RUSE metric, eliminating the need for a human-labeled dataset. We independently feed the original text and its summary into Mistral 7B, extracting the last hidden layer vectors from each inference. These vectors represent Mistral 7B's semantic understanding of the inputs. We then compute their cosine similarity (instead of the traditional MLP used in RUSE), normalizing the result to a 0-1 scale using: RUSE score = (cos-sim + 1)/2. This preserves semantic closeness (-1 for opposite, 1 for identical meanings) while eliminating dependence on embedding magnitude. Unlike BERTScore, which averages token-wise similarities and can be skewed by summary length, our approach directly computes the cosine similarity between full-text embeddings.(Figures 16, 17 allow us to compare our RUSE score with BERTScore)

4.2 Key findings from the evaluation phase

The evaluation phase reveals several key trends in the performance of our summarization models:

- The generated summaries often show low ROUGE scores (0–50%), which is unsurprising given that summarization relies heavily on rephrasing—making n-gram comparisons less effective. Another likely factor is the highly varied structure of the articles, resulting in minimal overlap which reflects our intended dataset design.
- ROUGE metrics are inherently biased by the reduction factor (the ratio of the original text to its summary), and BERTScore also exhibits a mild bias. Longer summaries tend to receive higher scores, making these metrics less reliable for evaluating summary quality. In contrast, LLM-as-judge and custom-RUSE evaluations remain unaffected by summary length, offering a more content-focused assessment.
- When projecting the custom-RUSE scores onto two dimensions using PCA, we observe that strong summaries naturally cluster together, making it easier to identify outliers and potential failure cases.
- ROUGE metrics predictably align, while BERTScore shows some agreement with ROUGE and moderate alignment with custom-RUSE. In contrast, LLM-as-judge stands apart, showing no correlation with the other metrics. However, interpreting its quality remains challenging, as deep learning models function as a "black box", making it unclear what aspects of a summary they prioritize.

These findings underscore the importance of using multiple evaluation metrics for ATS. While ROUGE is widely used, it provides significantly less insight than custom-RUSE, BERTScore, and LLM-as-judge, which better capture semantic quality. In the following subsections, we detail the individual evaluations of each model.

Table 2: Evaluation results (train/test) for various models and metrics

Model	Rouge 1	Rouge 2	Rouge L	BERT-score	LLM-as-judge	Custom RUSE
Large Instruct Models						
Mistral 24B Instruct 2501	**0.25**	**0.14**	**0.16**	**0.71**	**0.85**	**0.98**
Qwen2.5 32B Instruct	0.19	0.09	0.12	0.70	**0.85**	**0.98**
Fine-tuned Models						
prefix-tuned Llama-3.2 3B	0.15/0.14	0.09/0.09	0.11/0.10	0.58/0.57	0.72/0.74	0.78/0.77
LORA-tuned Llama-3.2 3B	0.25/0.25	0.14/0.14	0.16/0.16	0.71/0.71	**0.85/0.85**	**0.98/0.98**
paired curriculum Llama-3.2 3B	**0.44/0.44**	**0.43/0.43**	**0.44/0.43**	**0.90/0.89**	0.84/0.84	0.97/0.97
prefix-tuned Mistral 7B	0.25/0.25	0.15/0.16	0.18/0.18	0.71/0.71	**0.85/0.85**	**0.98/0.98**
LORA-tuned Mistral 7B	0.24/0.24	0.13/0.14	0.15/0.15	0.71/0.71	**0.85/0.85**	**0.98/0.98**
paired curriculum Mistral 7B	-/0.34	-/0.33	-/0.33	-/0.78	-/0.84	-/0.92

4.3 Analysis of our Large Language Models and of our baseline

Both models exhibit similar performance across all metrics, with Mistral 24B slightly outperforming Qwen2.5-32B. ROUGE scores for both range from 0 to 40% (Figures 5, 9), as discussed in Section 4.2. A key difference is that Qwen2.5-32B occasionally generates spurious summaries, as shown in Figure 12, whereas Mistral 24B does not. Manual inspection confirmed that both LLMs produced high-quality summaries, providing a strong gold standard for training our SLMs.

To effectively measure the impact of our fine-tuning process, we use the pretrained Mistral 7B and LLaMA-3.2 3B as baselines. A manual review of their generated summaries reveals frequent spurious outputs: some fail to generate text, others loop on words, and occasionally, they produce unrelated content (Figures 16, 21). This aligns with significantly lower evaluation scores (Figures 13, 18). Overall metrics are unexpectedly high, likely due to averaging instead of the median. Examining individual summaries reveals many low scores (e.g., RUSE < 0.5, indicating opposing meanings; see Figures 14, 19). .

4.4 Evaluating our fine-tuned Small Language Models

Our fine-tuned models perform remarkably well. They surprisingly do not exhibit signs of overfitting, despite the lack of early stopping in our fine-tuning process.

Comparing these models, we observe that prefix-tuned LLaMA-3.2 3B performs slightly worse than the others (Figure 25). However, LoRA-tuned LLaMA-3.2 3B, prefix-tuned Mistral 7B, and LoRA-tuned Mistral 7B achieve performance levels comparable to the gold standard (see table 2), demonstrating that our fine-tuning has effectively matched the summarization capabilities of the original LLMs. Additionally, prefix-tuning on Mistral 7B highlights the model's strong intrinsic summarization abilities (see reduction factor on

Figure 27), as the near-optimal prompt devised from the prefix-tuning process matches performance of the LLMs. This effect is less pronounced for LLaMA-3.2 3B, suggesting that Mistral's base model is inherently better suited for summarization. Paired curriculum learning yields promising results (Figures **??**, **??**), surpassing the gold standard on several metrics. This highlights the benefits of progressively increasing difficulty for fine-tuning and underscores the importance of pair or n-uplet learning in optimizing the model for target metrics. However we noticed some noise in the generated dataset (see Figures **??**, **??**). A manual evaluation of the generated summaries further confirms these findings: fine-tuned models produce consistent, coherent, and accurate summaries that remain easy to read, reinforcing the success of our fine-tuning approach.

For more details on the fine-tuning results, please refer to the appendix 5, where all our plots are provided.

5 Future direction and conclusion

In this work, we present a full methodology for fine-tuning SMLs for ATS. We begin by curating a diverse dataset of 5,000 French news articles sourced from La Presse which we preprocess following best practises. We then employ Mistral 24B and Qwen-2.5 32B Instruct to generate high-quality target summaries. Using this dataset, we fine-tune LLaMA-3.2 3B and Mistral 7B with LoRA, Prefix-Tuning, and Curriculum Learning, optimizing all processes through a Bayesian hyperparameter tuning phase. An extensive evaluation across multiple metrics demonstrates that all fine-tuned models, except LLaMA-3.2 3B with Prefix-Tuning, successfully match the summarization capabilities of the LLMs. These models generate coherent, accurate, and faithful summaries, a significant improvement compared to their pre-fine-tuned versions, which frequently produced spurious outputs and failed to grasp the task.

Throughout the project, we encountered several challenges. One issue was the naming of the files present in the dataset, as article titles contained French characters that repeatedly caused encoding errors, constantly disrupting the pipeline. Another challenge was managing numerous model runs, particularly on Polytechnique's computing infrastructure, which imposed strict disk space limitations per user and was prone to failures. The computing constraints made the project more difficult, but also more interesting as it highlighted the real-world struggles of working with limited computational resources, a factor often overlooked in research but crucial in practical applications.

We have many directions for possible future work, notably exploring other fine-tuning techniques such as adapters fine-tuning or reinforcement learning fine-tuning, or further combining techniques into a single fine-tuning process. Additionally, expanding the dataset with a larger sample of summaries of varying qualities for Curriculum Learning could improve training efficiency and is left for future research. Finally, fine-tuning smaller models from the Qwen family (Qwen2-0.5B or Qwen2-1.5B-Ins) would be an interesting direction, given their strong ATS performance throughout various benchmarks.

References

[1] Hadi Askari et al. *Assessing LLMs for Zero-shot Abstractive Summarization Through the Lens of Relevance Paraphrasing*. Feb. 1, 2025. DOI: 10.48550/arXiv.2406.03993. arXiv: 2406.03993[cs]. URL: http://arxiv.org/abs/2406.03993 (visited on 03/13/2025).

[2] James Bergstra et al. "Algorithms for Hyper-Parameter Optimization". In: *Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates, Inc., 2011. URL: https://papers.nips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html (visited on 03/13/2025).

[3] Alexis Conneau et al. *Unsupervised Cross-lingual Representation Learning at Scale*. version: 2. Apr. 8, 2020. DOI: 10.48550/arXiv.1911.02116. arXiv: 1911.02116[cs]. URL: http://arxiv.org/abs/1911.02116 (visited on 03/12/2025).

[4] Moussa Kamal Eddine et al. *AraBART: a Pretrained Arabic Sequence-to-Sequence Model for Abstractive Summarization*. Mar. 21, 2022. DOI: 10.48550/arXiv.2203.10945. arXiv: 2203.10945[cs]. URL: http://arxiv.org/abs/2203.10945 (visited on 03/12/2025).

[5] Colleen Gilhuly and Haleh Shahzad. *Consistency Evaluation of News Article Summaries Generated by Large (and Small) Language Models*. Feb. 28, 2025. DOI: 10.48550/arXiv.2502.20647. arXiv: 2502.20647[cs]. URL: http://arxiv.org/abs/2502.20647 (visited on 03/12/2025).

[6] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. Mar. 9, 2015. DOI: 10.48550/arXiv.1503.02531. arXiv: 1503.02531[stat]. URL: http://arxiv.org/abs/1503.02531 (visited on 03/12/2025).

[7] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. Oct. 16, 2021. DOI: 10.48550/arXiv.2106.09685. arXiv: 2106.09685[cs]. URL: http://arxiv.org/abs/2106.09685 (visited on 03/13/2025).

[8] Hanlei Jin et al. "A Comprehensive Survey on Process-Oriented Automatic Text Summarization with Exploration of LLM-Based Methods". In: (2024). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2403.02901. URL: https://arxiv.org/abs/2403.02901 (visited on 03/06/2025).

[9] Donghoon Kim et al. *Preserving Pre-trained Representation Space: On Effectiveness of Prefix-tuning for Large Multi-modal Models*. Oct. 29, 2024. DOI: 10.48550/arXiv.2411.00029. arXiv: 2411.00029[cs]. URL: http://arxiv.org/abs/2411.00029 (visited on 03/13/2025).

[10] Guillaume Lample and Alexis Conneau. *Cross-lingual Language Model Pretraining*. Jan. 22, 2019. DOI: 10.48550/arXiv.1901.07291. arXiv: 1901.07291[cs]. URL: http://arxiv.org/abs/1901.07291 (visited on 03/12/2025).

[11] Xiang Lisa Li and Percy Liang. "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. ACL-IJCNLP 2021. Ed. by Chengqing Zong et al. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. DOI: 10.18653/v1/2021.acl-long.353. URL: https://aclanthology.org/2021.acl-long.353/ (visited on 03/13/2025).

[12] 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/.

[13] Ying-Jia Lin et al. "Knowledge Distillation on Extractive Summarization". In: *2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. 2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE). Dec. 2020, pp. 71–76. DOI: 10.1109/AIKE48582.2020.00019. URL: https://ieeexplore.ieee.org/document/9355465/?arnumber=9355465 (visited on 03/12/2025).

[14] Sari Masri et al. "Transformer Models in Education: Summarizing Science Textbooks with AraBART, MT5, AraT5, and mBART". In: *Intelligent Systems, Blockchain, and Communication Technologies*. Ed. by Ahmed Abdelgawad, Akhtar Jamil, and Alaa Ali Hameed. Vol. 1268. Series Title: Lecture Notes in Networks and Systems. Cham: Springer Nature Switzerland, 2025, pp. 286–300. ISBN: 978-3-031-82376-3 978-3-031-82377-0. DOI: 10.1007/978-3-031-82377-0_25. URL: https://link.springer.com/10.1007/978-3-031-82377-0_25 (visited on 03/12/2025).

[15] Abdurrahman Odabaşı and Göksel Biricik. *Unraveling the Capabilities of Language Models in News Summarization*. Jan. 30, 2025. DOI: 10.48550/arXiv.2501.18128. arXiv: 2501.18128[cs]. URL: http://arxiv.org/abs/2501.18128 (visited on 03/13/2025).

[16] Telmo Pires, Eva Schlinger, and Dan Garrette. "How Multilingual is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. ACL 2019. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4996–5001. DOI: 10.18653/v1/P19-1493. URL: https://aclanthology.org/P19-1493/ (visited on 03/12/2025).

[17] Hassan Shakil, Ahmad Farooq, and Jugal Kalita. "Abstractive Text Summarization: State of the Art, Challenges, and Improvements". In: *Neurocomputing* 603 (Oct. 2024), p. 128255. ISSN: 09252312. DOI: 10.1016/j.neucom.2024.128255. arXiv: 2409.02413[cs]. URL: http://arxiv.org/abs/2409.02413 (visited on 03/13/2025).

[18] Alexander Spangher et al. "NewsEdits: A News Article Revision Dataset and a Novel Document-Level Reasoning Challenge". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. NAACL-HLT 2022. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 127–157. DOI: 10.18653/v1/2022.naacl-main.10. URL: https://aclanthology.org/2022.naacl-main.10/ (visited on 03/13/2025).

[19] Chenxi Whitehouse et al. *Low-Rank Adaptation for Multilingual Summarization: An Empirical Study*. Mar. 31, 2024. DOI: 10.48550/arXiv.2311.08572. arXiv: 2311.08572[cs]. URL: http://arxiv.org/abs/2311.08572 (visited on 03/13/2025).

[20] Borui Xu et al. *Evaluating Small Language Models for News Summarization: Implications and Factors Influencing Performance*. Feb. 11, 2025. DOI: 10.48550/arXiv.2502.00641. arXiv: 2502.00641[cs]. URL: http://arxiv.org/abs/2502.00641 (visited on 03/12/2025).

[21] Tianyi Zhang et al. *Benchmarking Large Language Models for News Summarization*. Jan. 31, 2023. DOI: 10.48550/arXiv.2301.13848. arXiv: 2301.13848[cs]. URL: http://arxiv.org/abs/2301.13848 (visited on 03/12/2025).

[22] Tianyi Zhang et al. *BERTScore: Evaluating Text Generation with BERT*. 2020. arXiv: 1904.09675 [cs.CL]. URL: https://arxiv.org/abs/1904.09675.

[23] Lulu Zhao et al. *Domain-Oriented Prefix-Tuning: Towards Efficient and Generalizable Fine-tuning for Zero-Shot Dialogue Summarization*. Apr. 9, 2022. DOI: 10.48550/arXiv.2204.04362. arXiv: 2204.04362[cs]. URL: http://arxiv.org/abs/2204.04362 (visited on 03/13/2025).

Appendices

A Curating a dataset

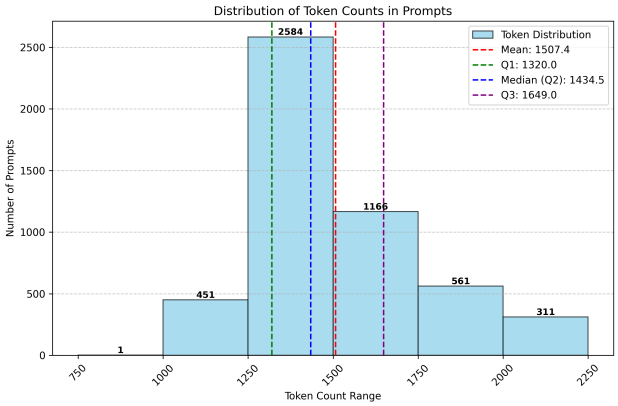

Figure 1: Token-count distribution of the dataset after filtering

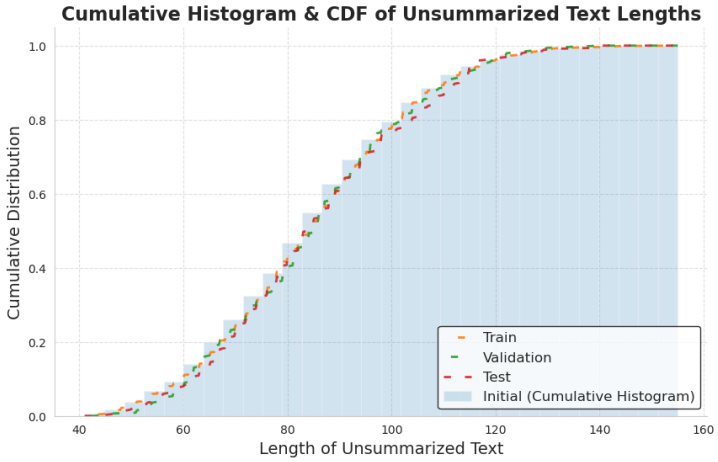

Figure 2: Split of the dataset in 3 subdataset respecting the initial distribution of documents (base on the length)

B Fine-Tuning the Small Language Models

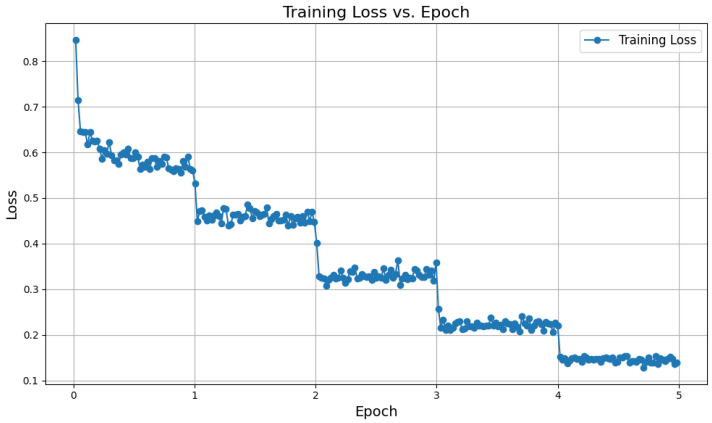

Figure 3: Training loss against the epoch for LoRA fine-tune of Llama-3.2 3B

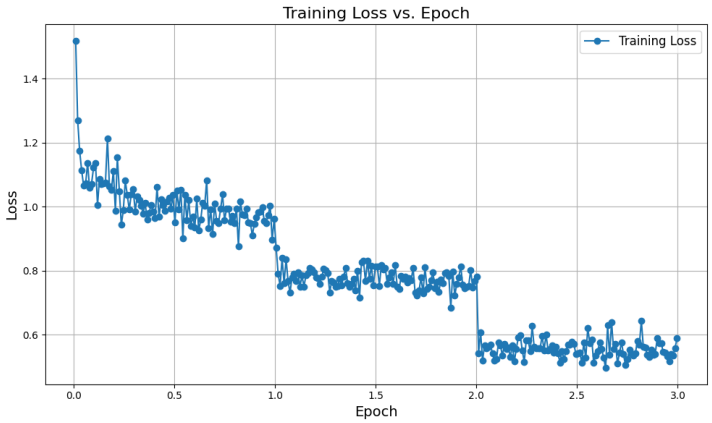

Figure 4: Training loss against the epoch for LoRA fine-tune of Mistral 7B

C Evaluation of our models

Veuillez évaluer la qualité du résumé ci-dessous en lui attribuant une note finale comprise entre 0.00 et 10.00. Cette note doit refléter l'évaluation globale du résumé selon les critères suivants :

Fidélité: Le résumé restitue-t-il correctement et complètement les informations essentielles du texte original ?
Concision: Le résumé présente-t-il l'information de manière synthétique sans détails superflus ?
Clarté: Le résumé est-il formulé de manière claire, compréhensible et structurée ?
Contexte: Le résumé est-il correctement contextualisé ?

0.00 signifie « très mauvais »
10.00 signifie « excellent »
Répondez uniquement avec un nombre décimal à deux chiffres après la virgule (par exemple, 7.58).

Texte original:
{initial_text}

Résumé:
{summary} Note (entre 0.00 et 10.00):

Prompt processed by LLM-as-judge Mistral 7B to rate the generated summaries

D Dataset generation Mistral 24B Instruct 2501

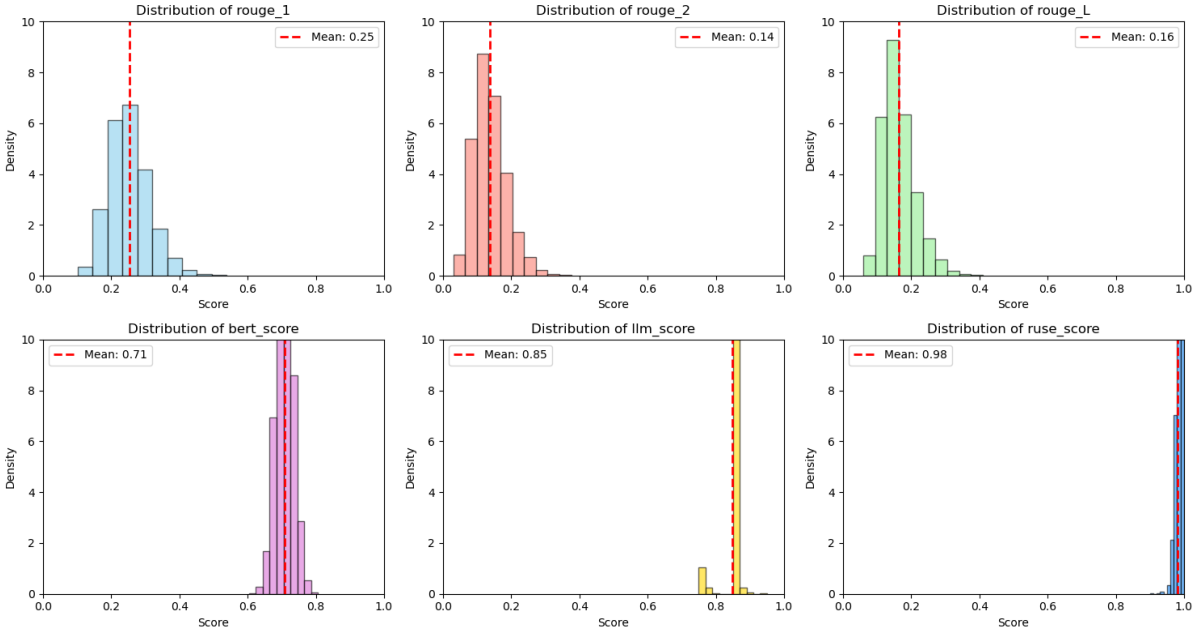

Figure 5: Distribution of the score metrics across the dataset (Mistral 24B Instruct).

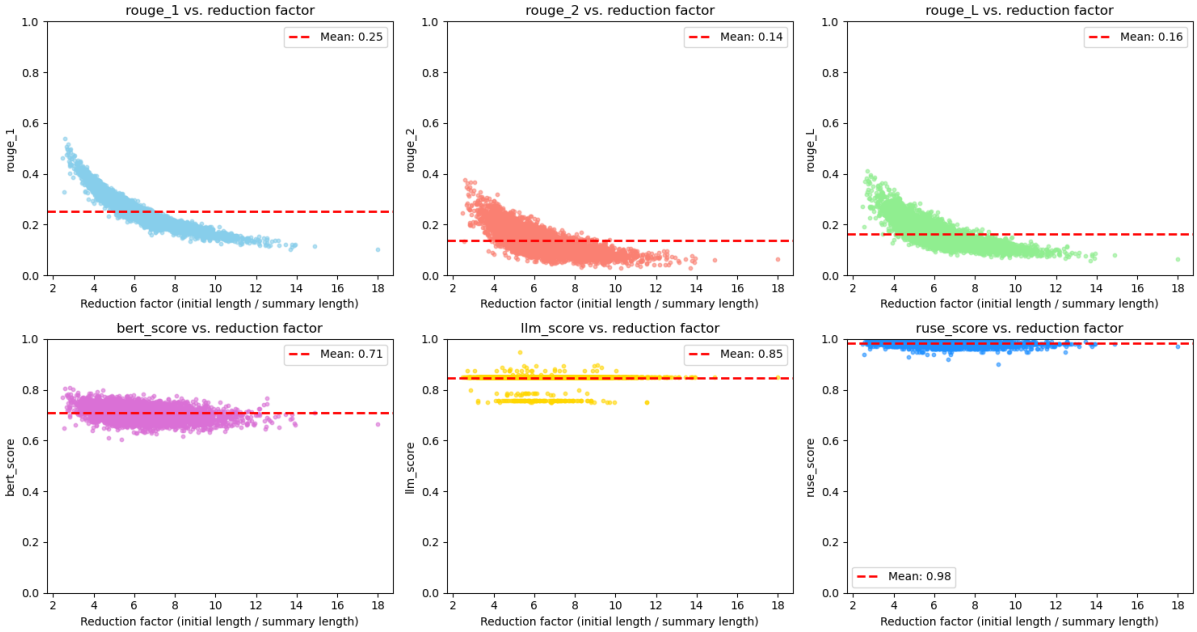

Figure 6: Relationship between score metrics and the reduction factor (Mistral 24B Instruct).

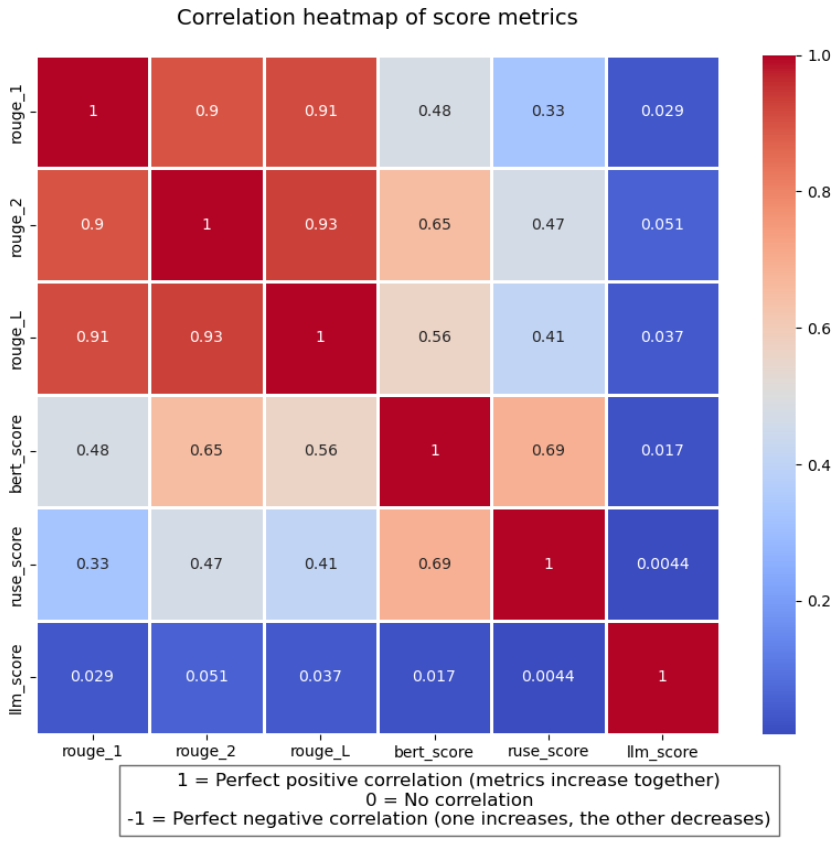

Figure 7: Heatmap illustrating the correlation between different score metrics (Mistral 24B Instruct).

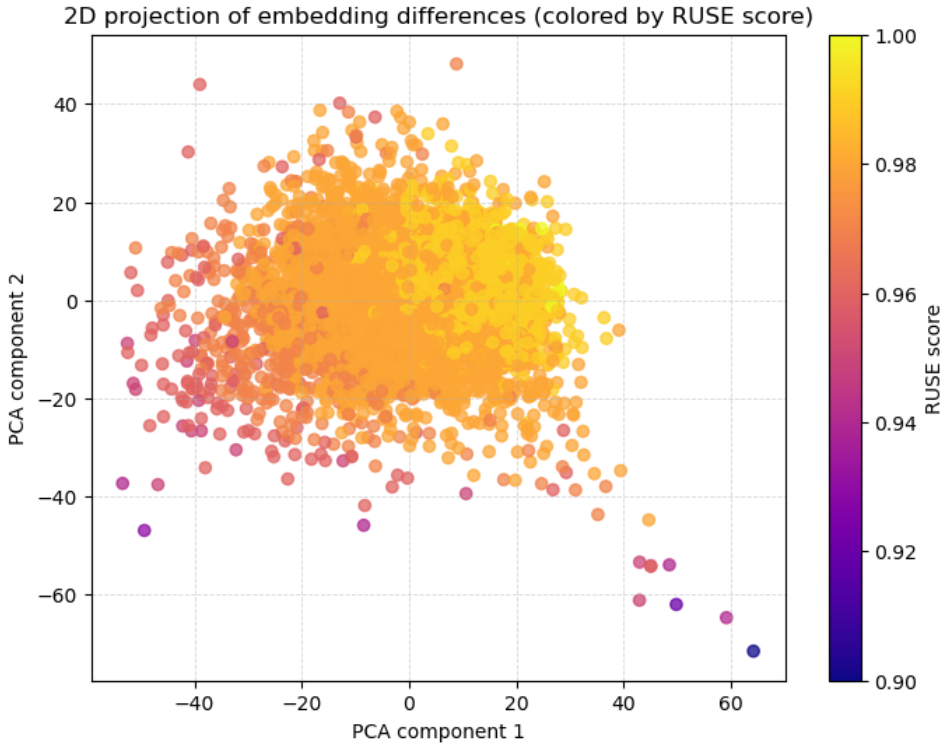

Figure 8: Representation of summarization operation embeddings based on the RUSE metric (Mistral 24B Instruct).

E Dataset generation Qwen2.5 32B Instruct

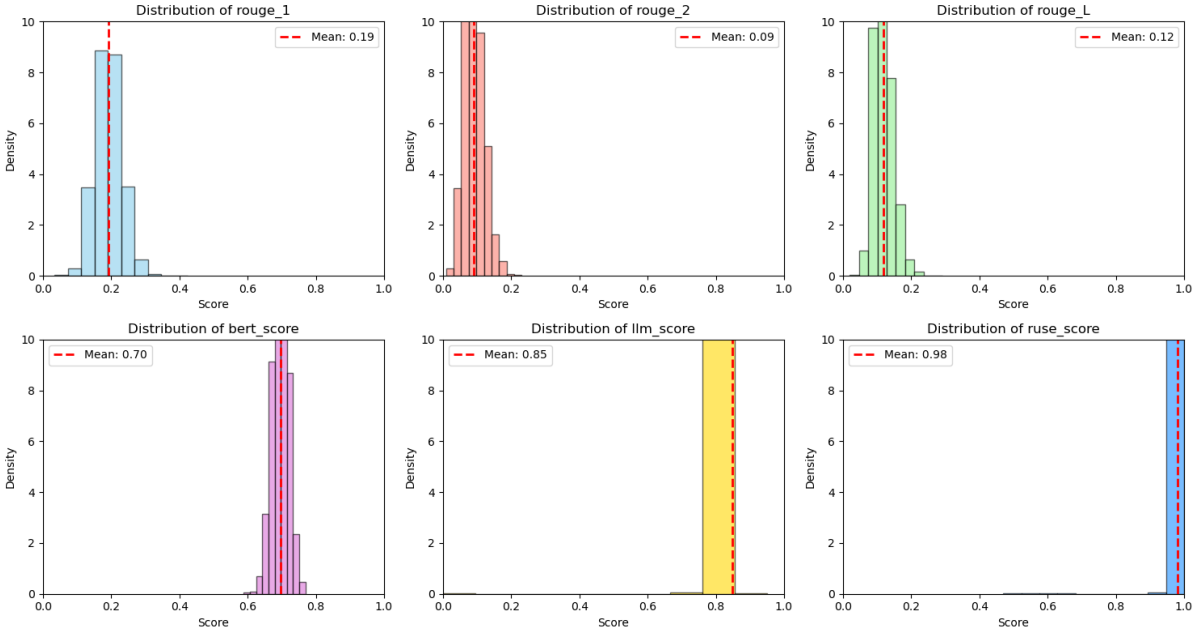

Figure 9: Distribution of the score metrics across the dataset (Qwen2.5 32B Instruct).

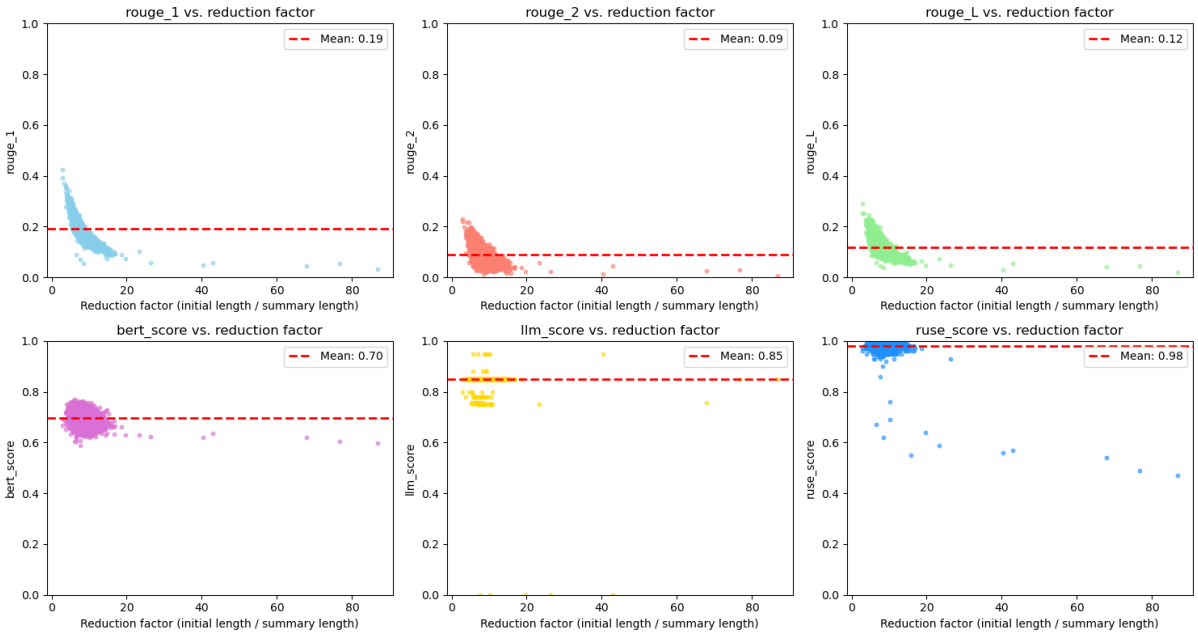

Figure 10: Relationship between score metrics and the reduction factor (Qwen2.5 32B Instruct).

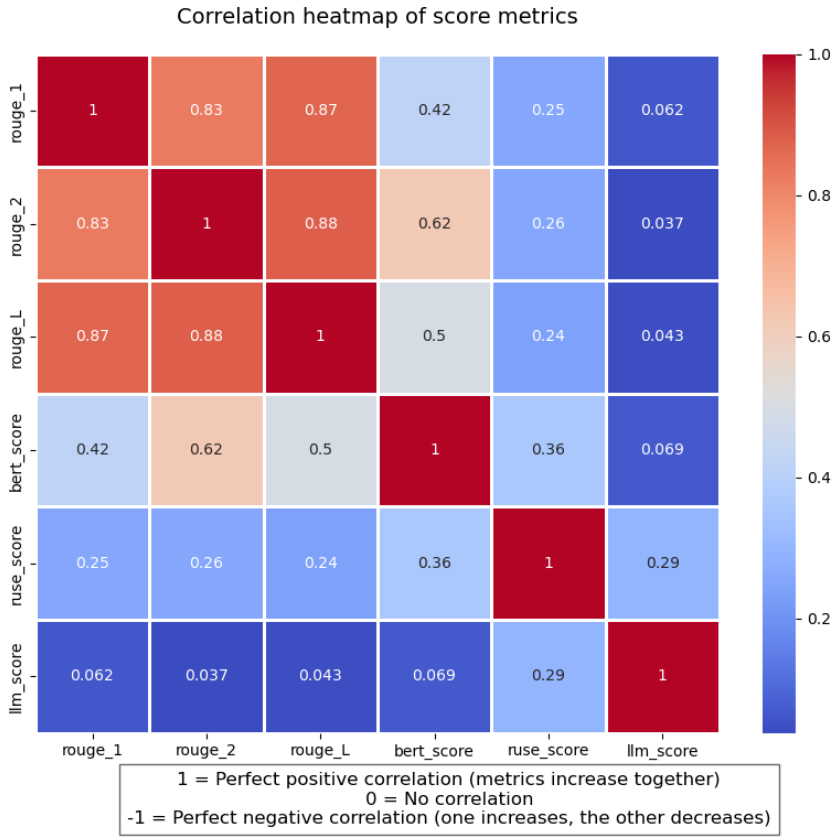

Figure 11: Heatmap illustrating the correlation between different score metrics (Qwen2.5 32B Instruct).

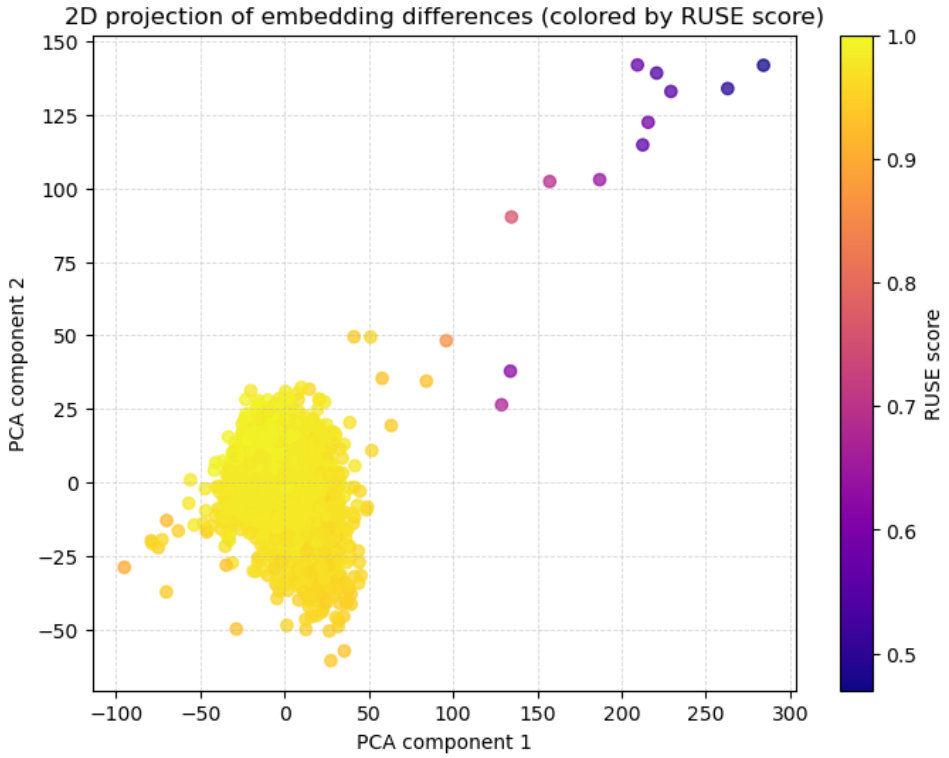

Figure 12: Representation of summarization operation embeddings based on the RUSE metric (Qwen2.5 32B Instruct).

F Baseline Llama3.2 3B

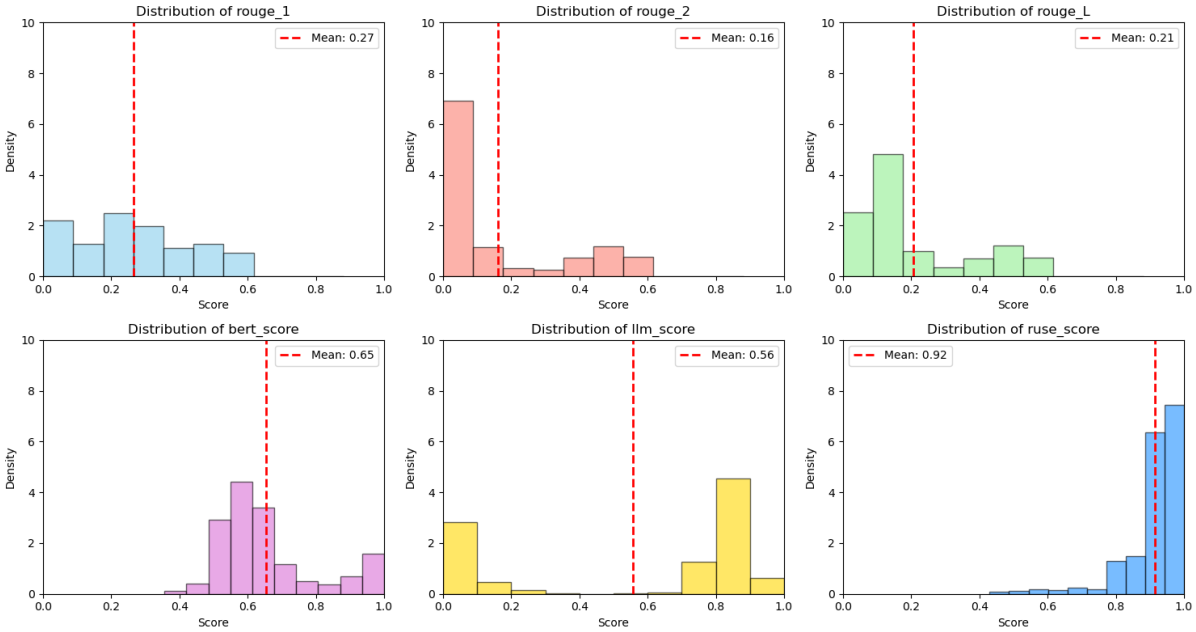

Figure 13: Distribution of the score metrics across the dataset (Llama3.2 3B - baseline).

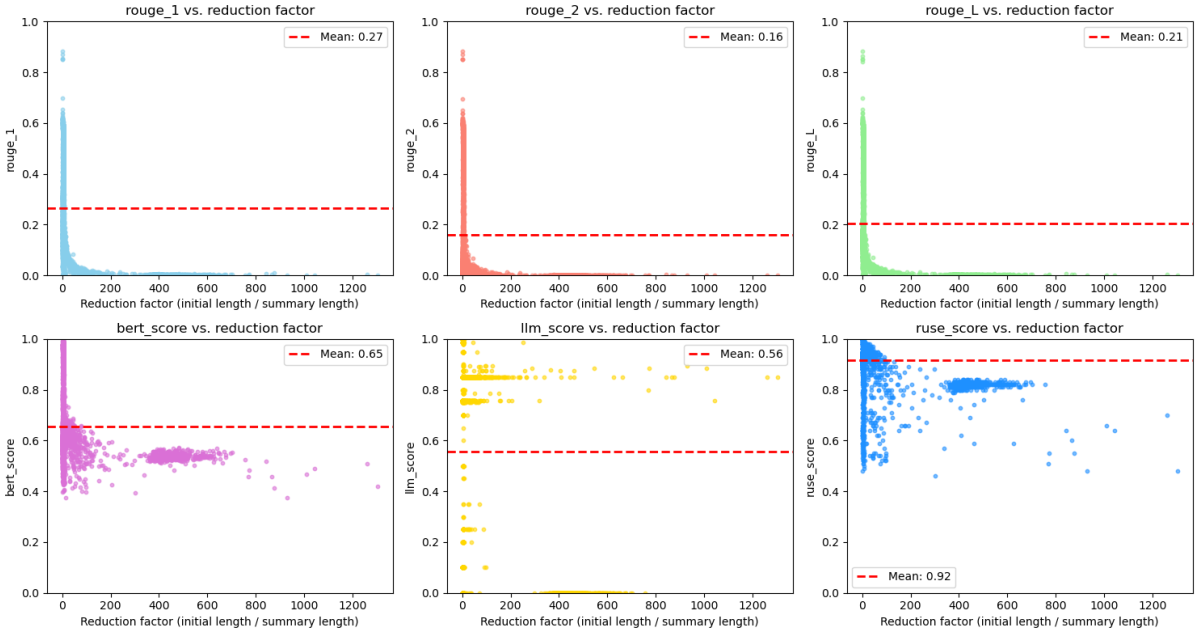

Figure 14: Relationship between score metrics and the reduction factor (Llama3.2 3B - baseline).

Figure 15: Heatmap illustrating the correlation between different score metrics (Llama3.2 3B - baseline).

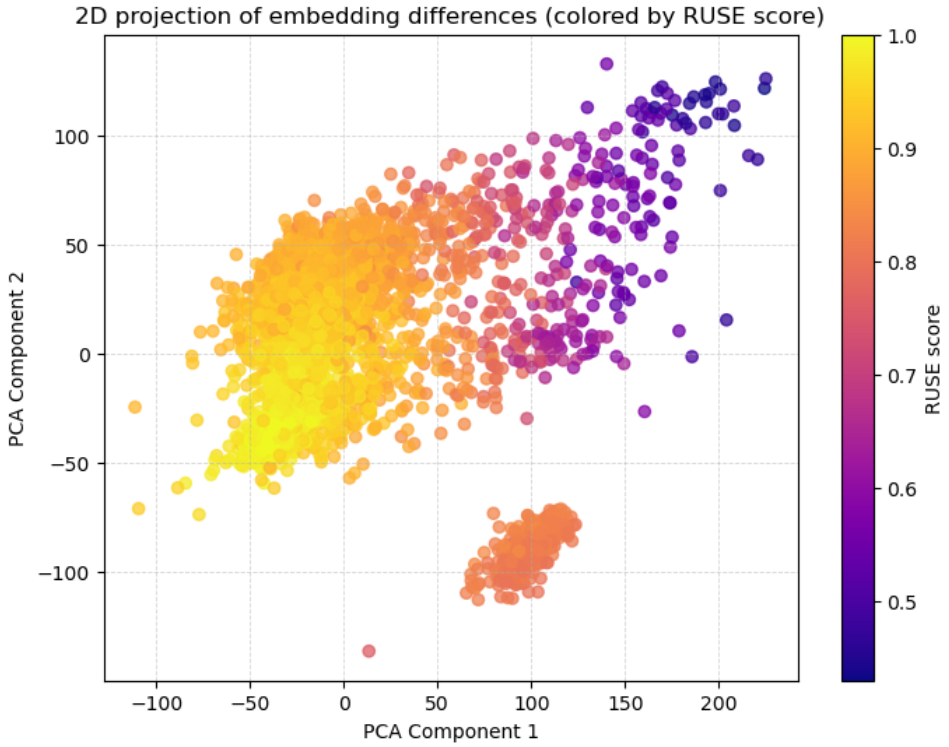

Figure 16: Representation of summarization operation embeddings based on the RUSE metric (Llama3.2 3B - baseline).

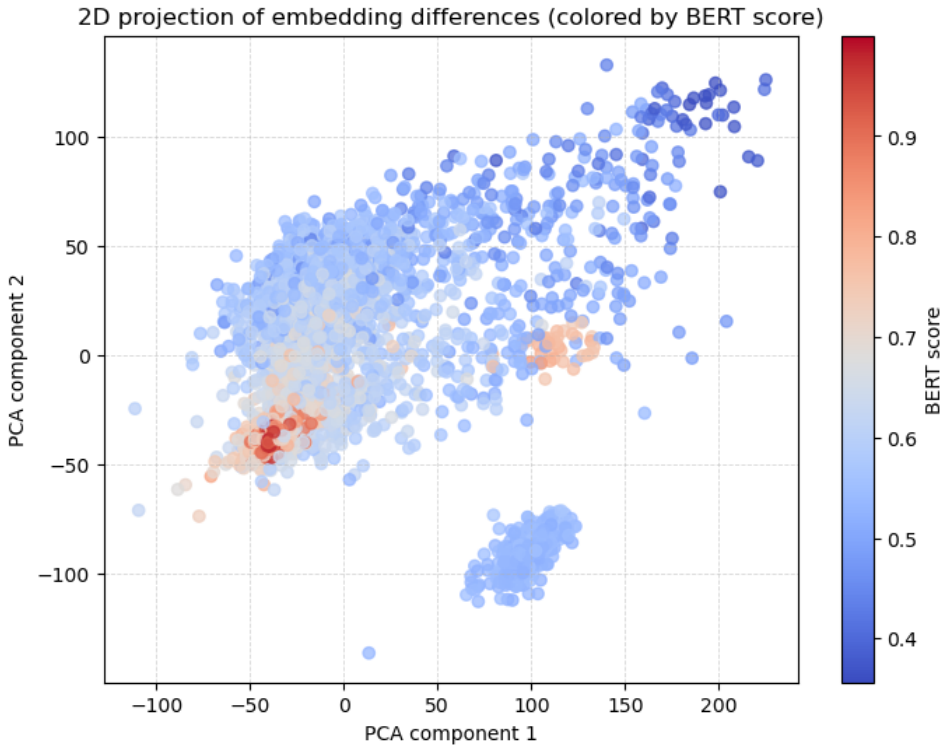

Figure 17: Representation of summarization operation embeddings based on the RUSE metric compared with BERTScore (Llama3.2 3B - baseline).

G Baseline Mistral 7B v0.1

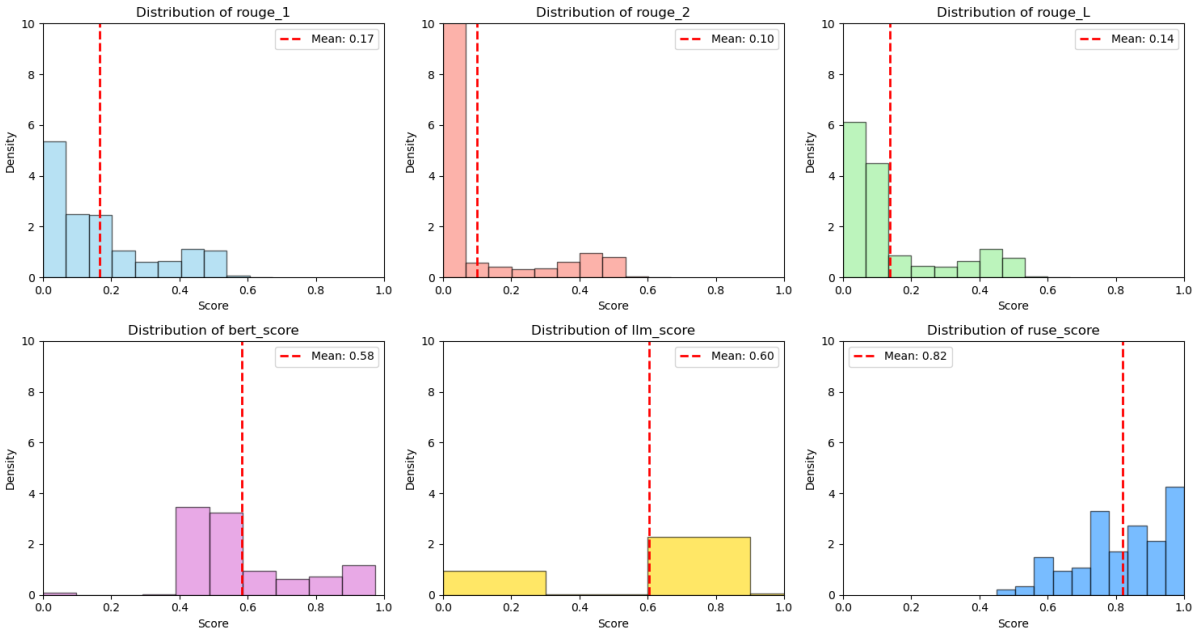

Figure 18: Distribution of the score metrics across the dataset (Mistral 7B v0.1 - baseline).

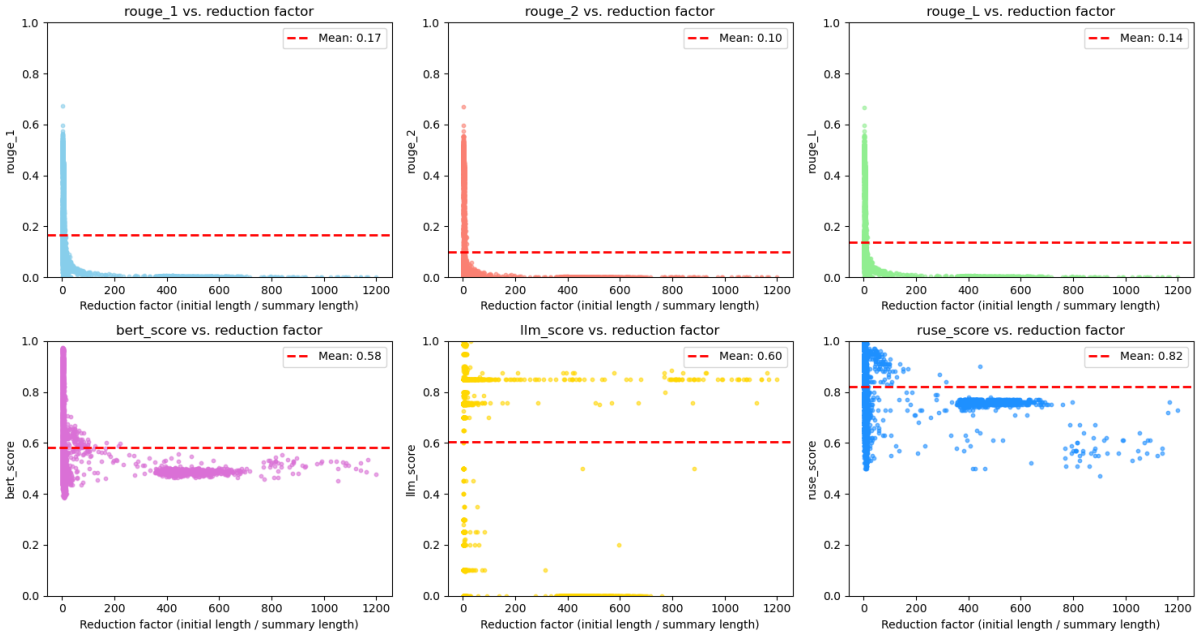

Figure 19: Relationship between score metrics and the reduction factor (Mistral 7B v0.1 - baseline).

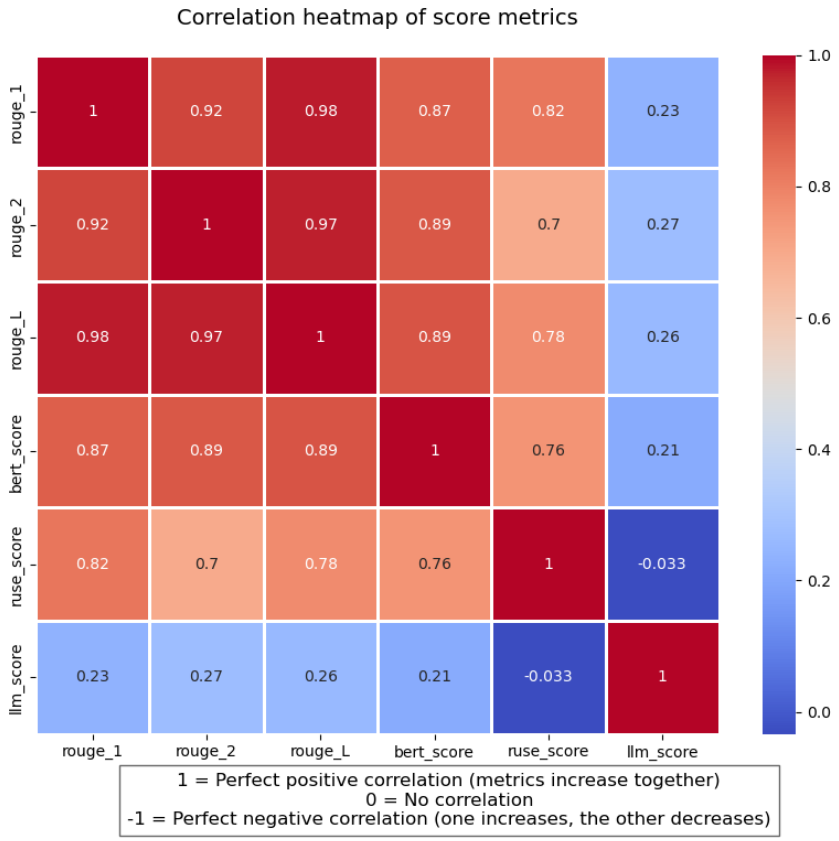

Figure 20: Heatmap illustrating the correlation between different score metrics (Mistral 7B v0.1 - baseline).

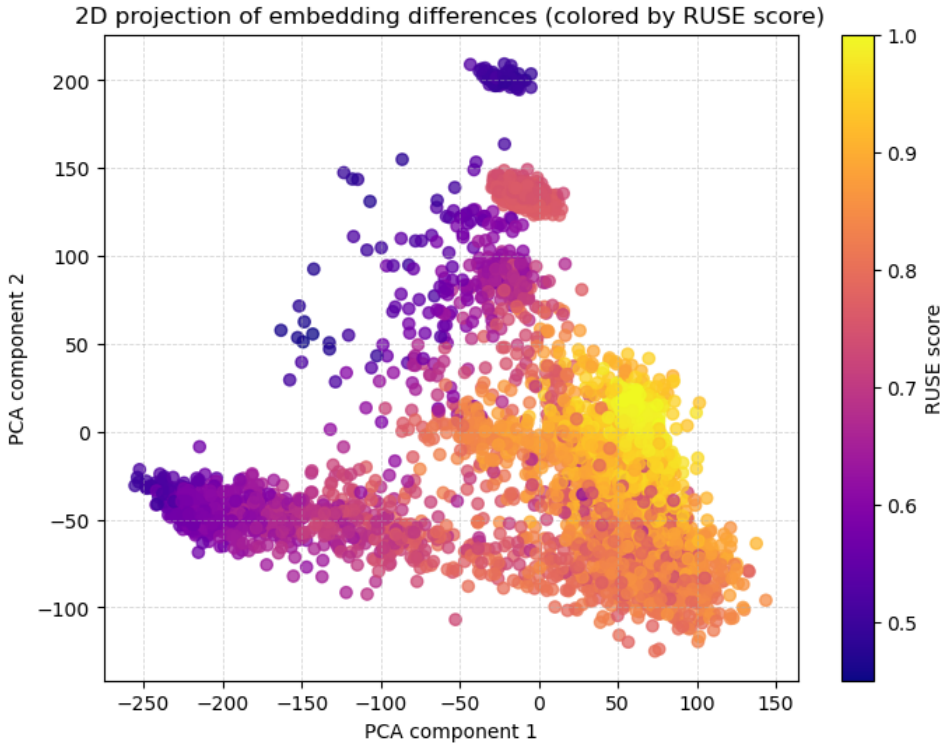

Figure 21: Representation of summarization operation embeddings based on the RUSE metric (Mistral 7B v0.1 - baseline).

H Prefix tuning Llama3.2 3B

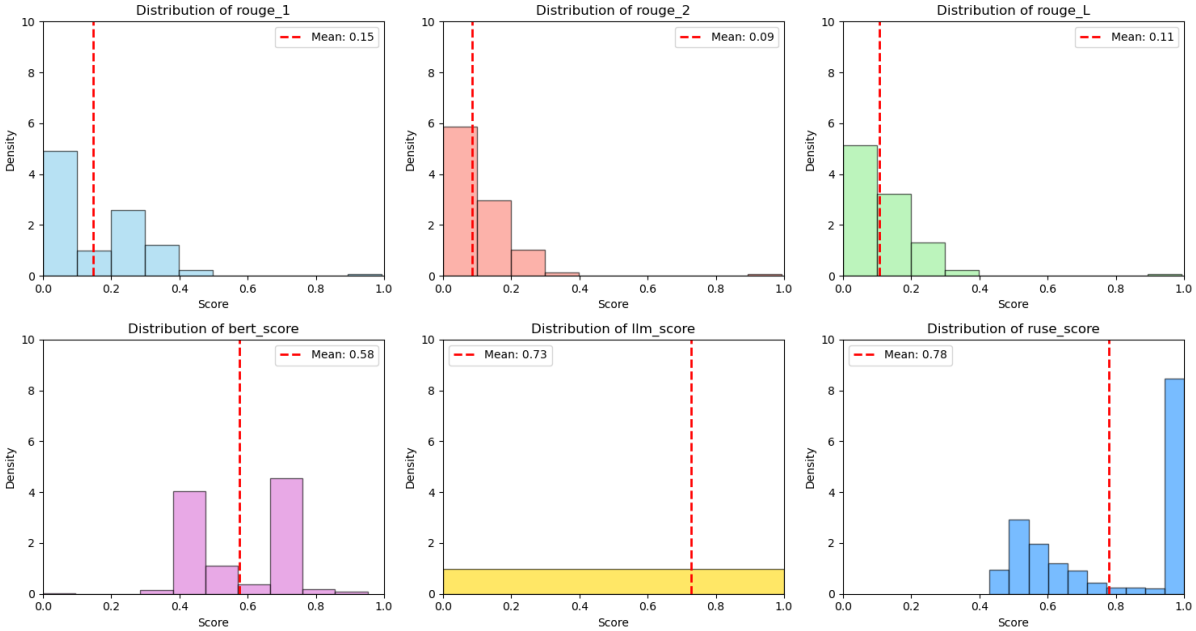

Figure 22: Distribution of the score metrics across the dataset (Llama 3.2 3B - prefix tuning).

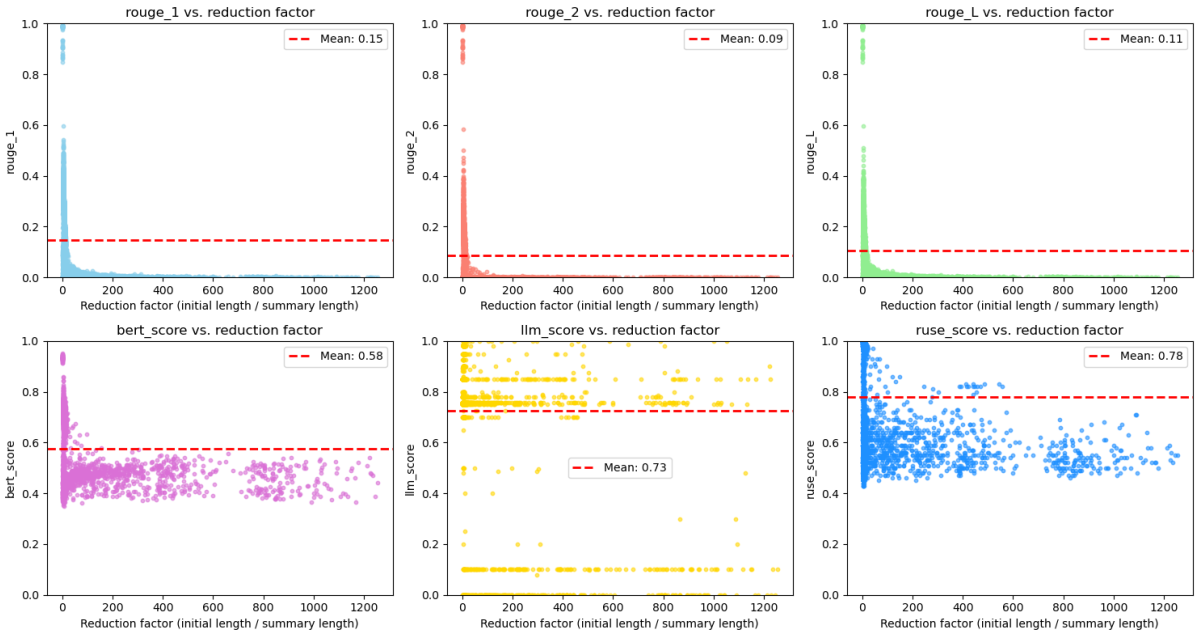

Figure 23: Relationship between score metrics and the reduction factor (Llama 3.2 3B - prefix tuning).

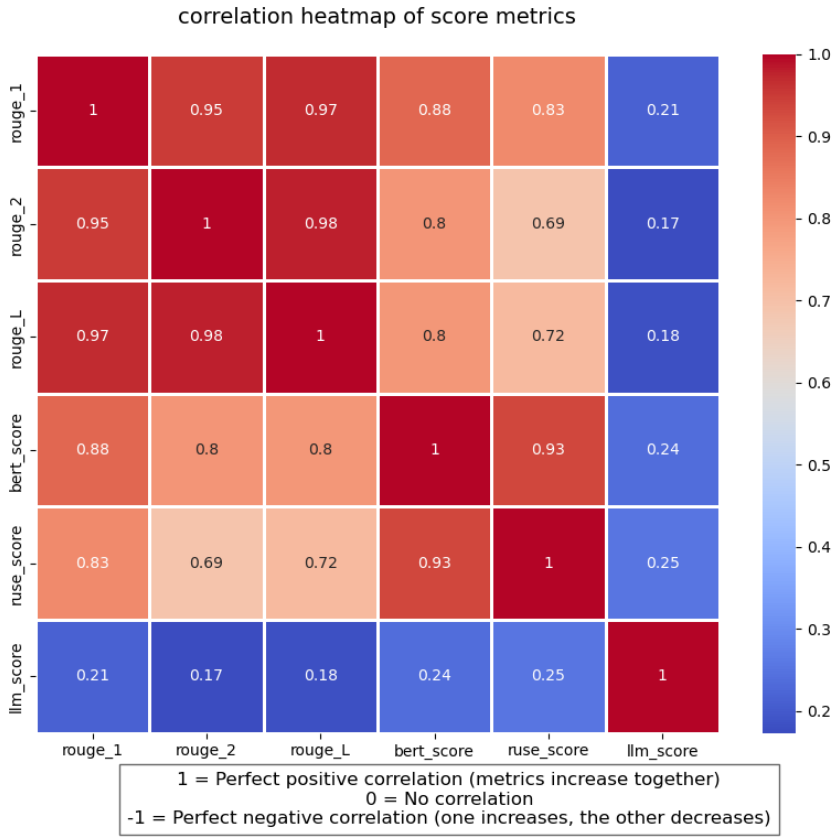

Figure 24: Heatmap illustrating the correlation between different score metrics (Llama 3.2 3B - prefix tuning).

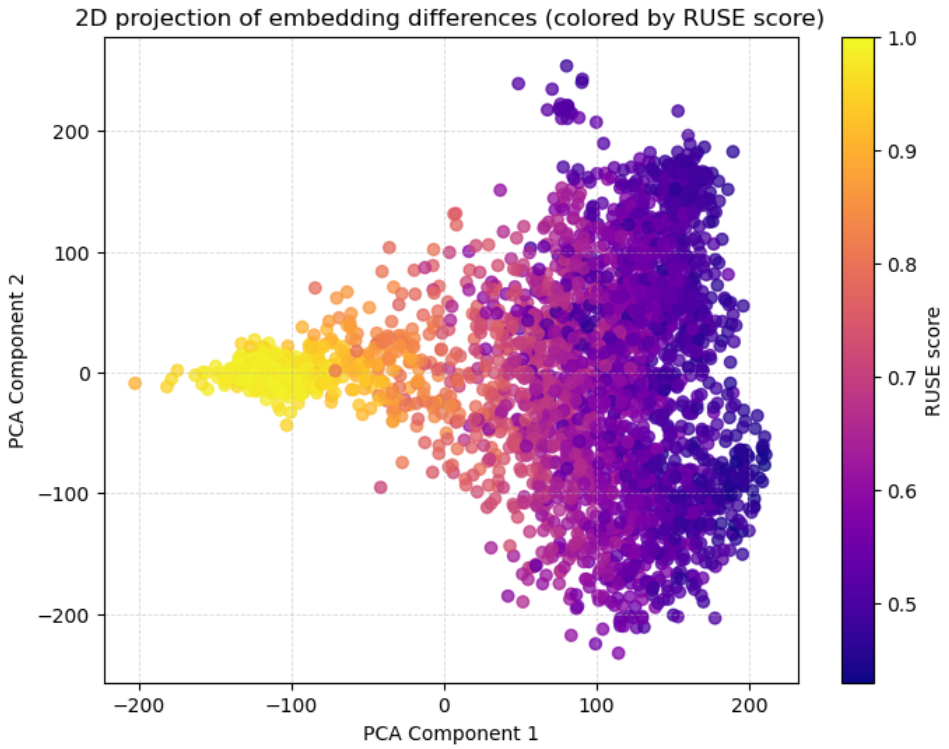

Figure 25: Representation of summarization operation embeddings based on the RUSE metric (Llama 3.2 3B - prefix tuning).

I Prefix tuning Mistral 7B v0.1

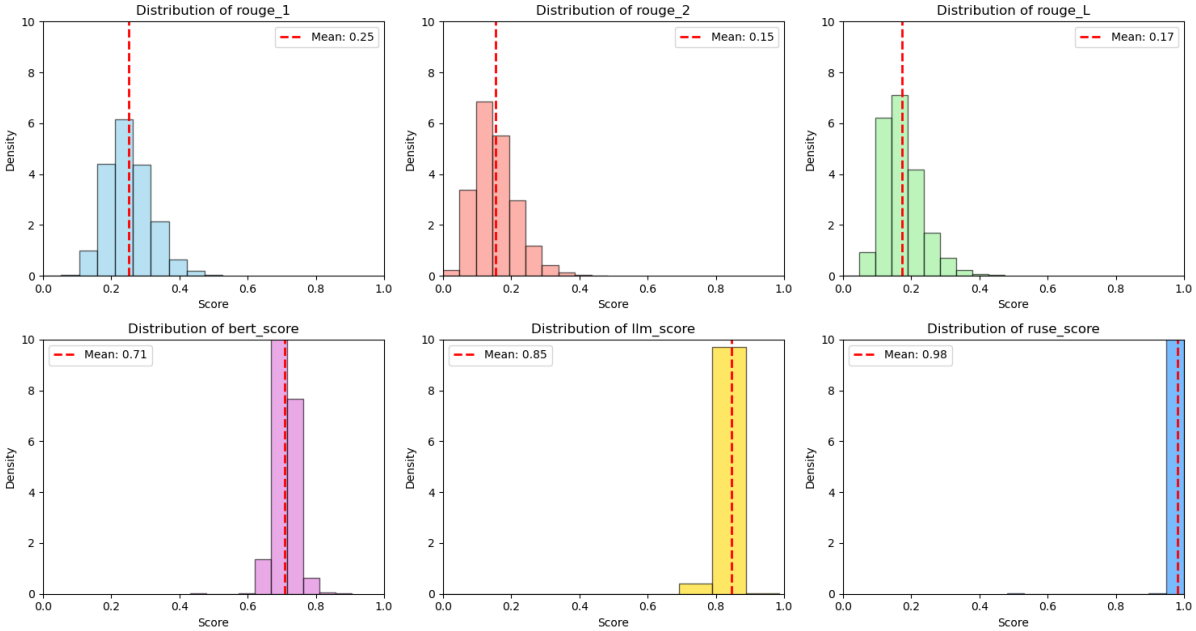

Figure 26: Distribution of the score metrics across the dataset (Mistral 7B v0.1 - prefix tuning).

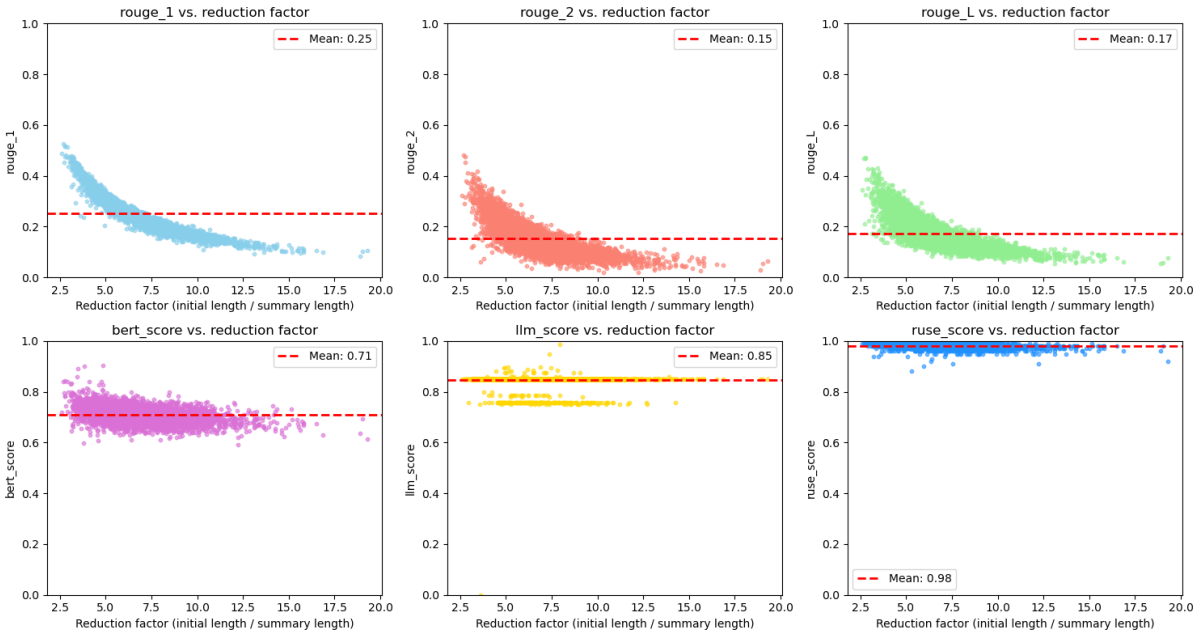

Figure 27: Relationship between score metrics and the reduction factor (Mistral 7B v0.1 - prefix tuning).

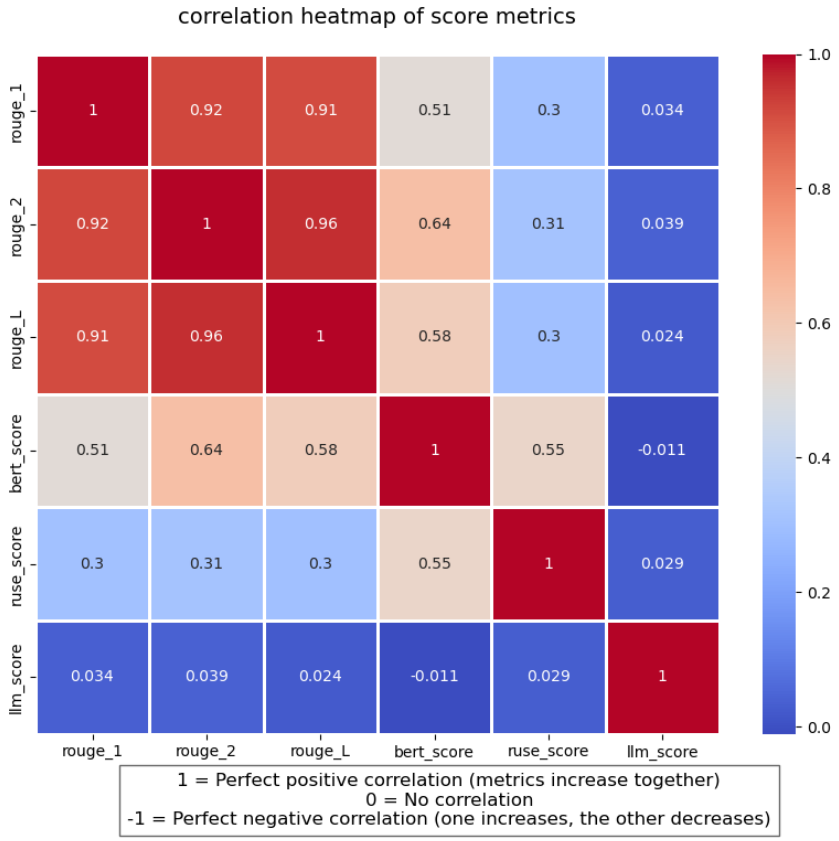

Figure 28: Heatmap illustrating the correlation between different score metrics (Mistral 7B v0.1 - prefix tuning).

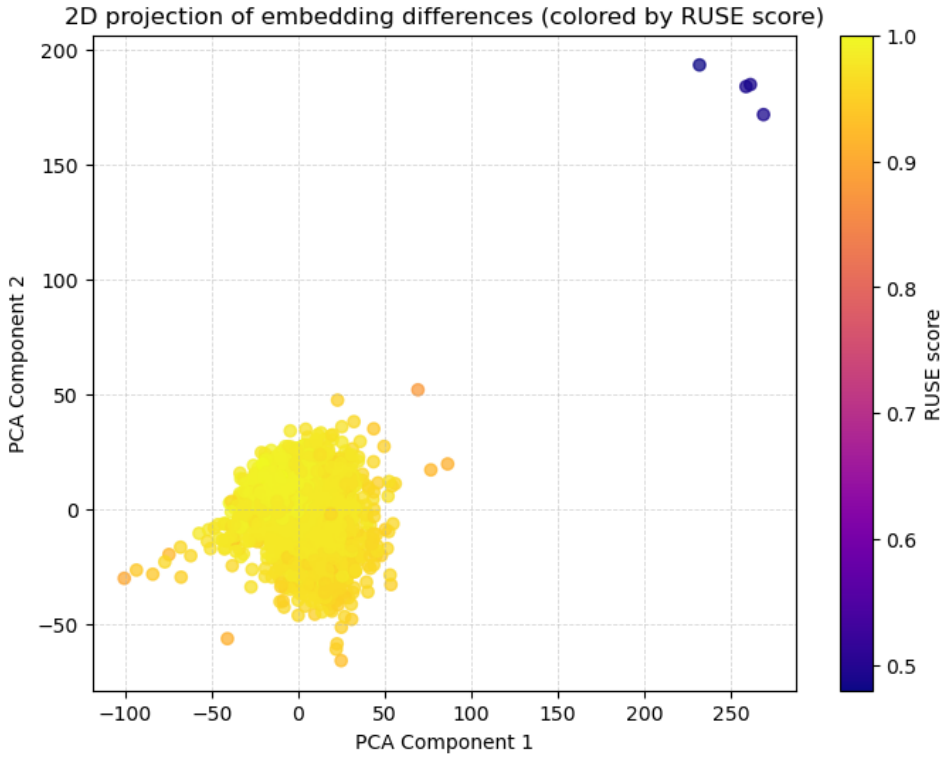

Figure 29: Representation of summarization operation embeddings based on the RUSE metric (Mistral 7B v0.1 - prefix tuning).

J LoRA tuning Llama3.2 3B

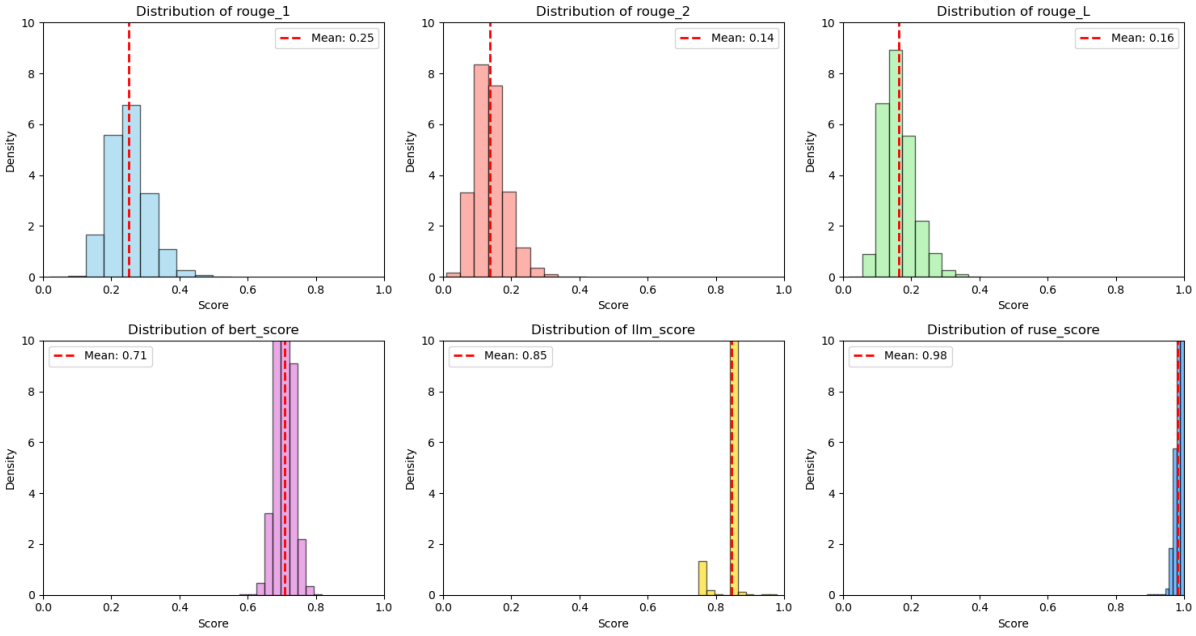

Figure 30: Distribution of the score metrics across the dataset (Llama 3.2 3B - LoRA tuning).

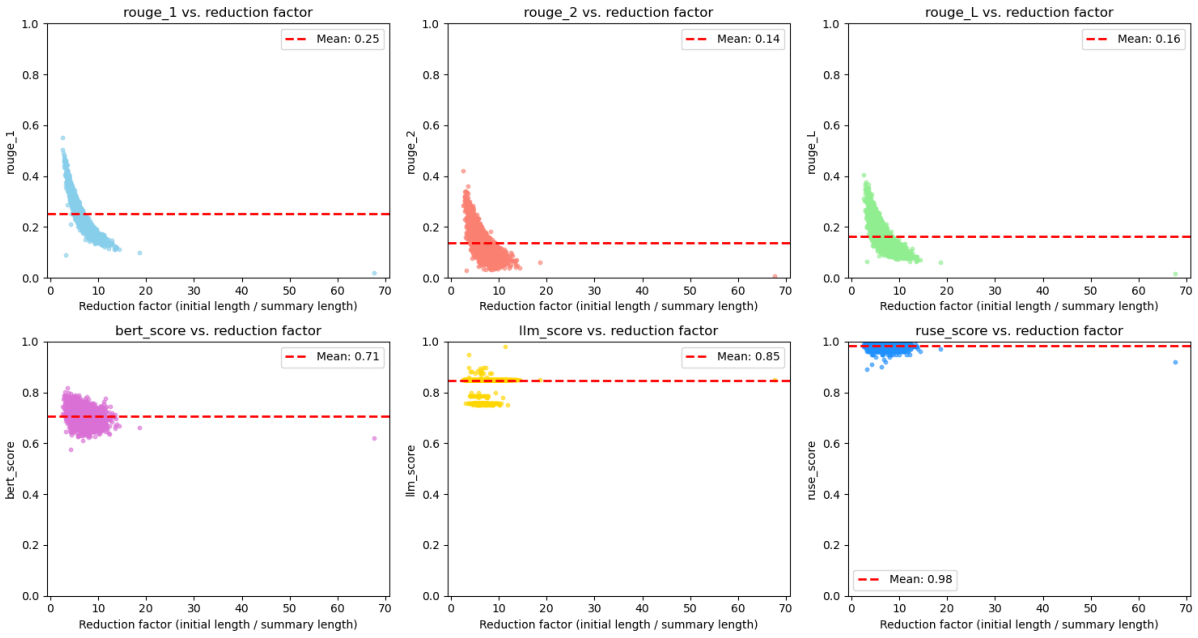

Figure 31: Relationship between score metrics and the reduction factor (Llama 3.2 3B - LoRA tuning).

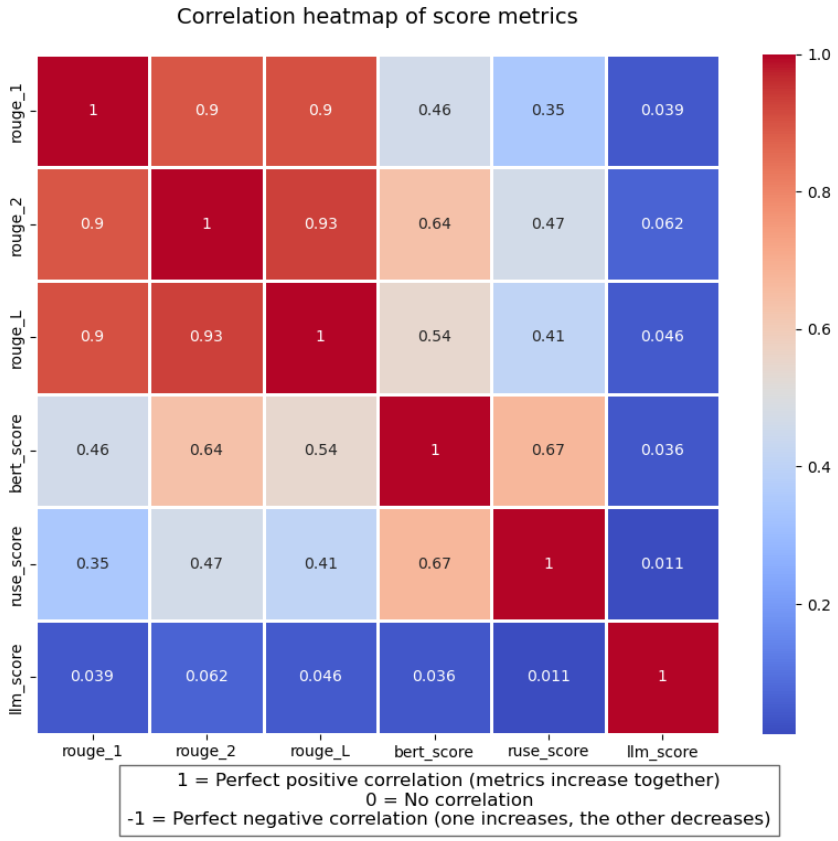

Figure 32: Heatmap illustrating the correlation between different score metrics (Llama 3.2 3B - LoRA tuning).

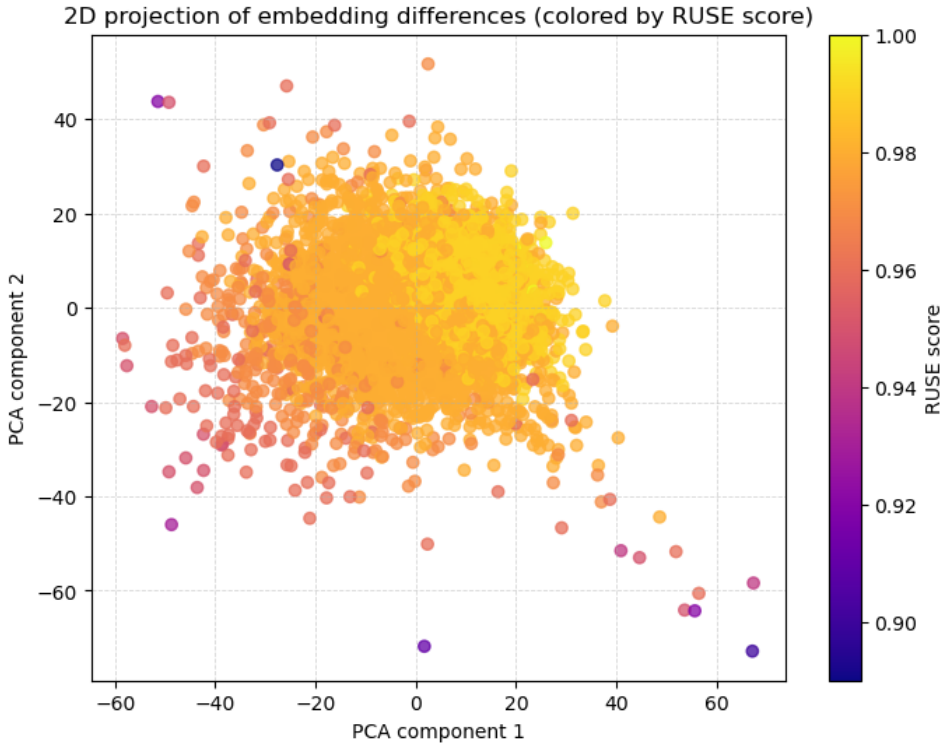

Figure 33: Representation of summarization operation embeddings based on the RUSE metric (Llama 3.2 3B - LoRA tuning).

K LoRA tuning Mistral 7B v0.1

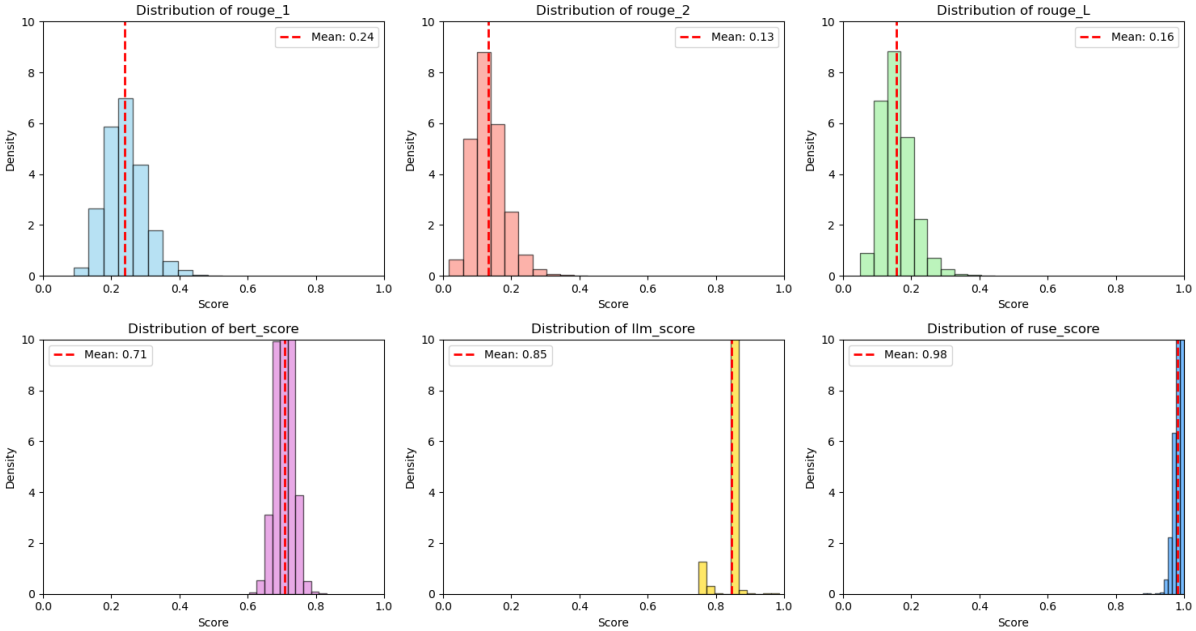

Figure 34: Distribution of the score metrics across the dataset (Mistral 7B v0.1 - LoRA tuning).