Journal of the Brazilian Computer Society, 2025, 31:1, doi: 10.5753/jbcs.2025.5811

Abstractive Summarization with LLMs for Texts in Brazilian Portuguese

Hugo A. P. G. de Camargo [**University of São Paulo** | ***hgpadovani@gmail.com***]
Pedro Henrique Paiola [**São Paulo State University** | ***pedro.paiola@unesp.br***]
Gabriel Lino Garcia [**São Paulo State University** | ***gabriel.lino@unesp.br***]
João Paulo Papa [**São Paulo State University** | ***joao.papa@unesp.br***]

University of São Paulo, Sciences, Mathematics and Computing Institute, Avenida Trabalhador São-carlense, 400 - Centro CEP: 13566-590 - São Carlos - SP

Received: 01 April 2025 • **Accepted:** 17 July 2025 • **Published:** 14 October 2025

Abstract This study aims to compare large language models (LLMs) in the task of text summarization for Portuguese-language texts. A dataset of 8, 116 samples was used, containing the original texts and their corresponding reference summaries. Initially, an experiment was conducted comparing three different prompts using zero-shot, one-shot, and few-shot techniques, processing 100 samples for four out of the six models (those that accept instructions as part of their input). The goal of this preliminary experiment was to determine an optimal prompt for conducting the full-scale experiment. After selecting the prompt, a second experiment was performed, running all six models on the 8, 116 samples and evaluating summarization quality using metrics such as BLEU and ROUGE, as well as Compression Rate and Inference Time for the generated summaries. Finally, an experiment was conducted to analyze the impact of 4-bit and 8-bit quantization, assessing how these different configurations affect the generated summaries, evaluation metrics, Compression Rate, and Inference Time.

Keywords: Large Language Models, Abstractive Summarization, Machine Learning, Generative AI, Natural Language Processing

1 Introduction

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP), demonstrating state-of-the-art performance in text generation, translation, and summarization. The growing volume of digital content has heightened the need for efficient information processing, making text summarization an increasingly critical task. Among summarization techniques, abstractive summarization is particularly valuable, as it generates novel text rather than extracting sentences from the input.

LLMs have shown promise in both extractive and abstractive summarization tasks. Extractive summarization involves selecting key sentences from the original text, whereas abstractive summarization generates novel sentences that capture the core ideas. A comprehensive survey by [Zhang *et al.*, 2024] provides an overview of the evolution of text summarization techniques, highlighting the transition from statistical methods to the integration of LLMs. Recent studies have demonstrated the efficacy of LLMs in summarization across various domains. For instance, [Basyal and Sanghvi, 2023] conducted a comparative study on text summarization using models such as MPT-7b-instruct, Falcon-7b-instruct, and OpenAI's ChatGPT, highlighting the strengths and limitations of each in generating summaries.

In specialized fields, LLMs have also shown potential. For example, [Van Veen *et al.*, 2023] demonstrated that adapted LLMs can outperform medical experts in clinical text summarization tasks, suggesting their potential in alleviating documentation burdens in healthcare settings. However, challenges remain in ensuring the factual consistency and reliability of summaries generated by LLMs. In ongoing research, the need for robust evaluation metrics to assess the precision of LLM-generated summaries is emphasized, particularly in critical domains such as healthcare.

Despite their potential, LLMs' performance in abstractive summarization is highly dependent on prompt design. Prompt engineering—the practice of crafting effective instructions—directly influences summary quality, inference speed, and computational efficiency. However, the impact of different prompt structures remains underexplored, necessitating a systematic investigation of their effects on summarization outcomes.

In the Portuguese language context, the development of summarization systems has advanced with the creation of dedicated datasets and benchmarks. The CSTNews corpus [Cardoso *et al.*, 2011] is one of the earliest manually annotated resources for multi-document summarization in Brazilian Portuguese. More recently, [Fonseca *et al.*, 2016] introduced the Summ-it++ dataset which is based on the work proposed by [Collovini *et al.*, 2007], the idea of this new large-scale corpus composed of journalistic texts and human-written abstractive summaries, specifically designed to evaluate summarization models in Portuguese. It is also possible to create domain-specific corpora. For instance, RulingBR [de Vargas Feijó and Moreira, 2018] focuses on summarizing legal texts in Portuguese, comprising approximately 10, 000 decisions from the Brazilian Supreme Federal Court.

This study evaluates three distinct prompt engineering strategies for abstractive summarization. Using a Portuguese dataset of approximately $8,000$ samples, we assess summary quality through BLEU [Papineni *et al.*, 2002] and ROUGE [Lin, 2004] scores, alongside inference time. Additionally, we examine the effects of model quantization at 4- and 8-bit levels to balance performance and efficiency. Our findings provide insights into optimizing prompt strategies for improved summarization quality with reduced computational cost.

The remainder of this paper is organized as follows: Section 2 reviews related work in summarization and prompt engineering. Section 3 details the methodology, including the selection of the data set, the models, and the evaluation metrics. Section 4 presents experimental results and discussion. Finally, Section 5 concludes the study and outlines future research directions.

2 Related Works

2.1 Introduction to Text Summarization

Text summarization is a crucial area in natural language processing and artificial intelligence (AI), playing a significant role in organizing and condensing large volumes of information. Its importance extends to both academia and industry, enabling efficient information retrieval and optimized reading time. Summarization can be broadly categorized into extractive and abstractive approaches, each with distinct characteristics and challenges [Nenkova and McKeown, 2011].

2.2 Extractive Summarization

Extractive summarization involves selecting and concatenating representative segments, typically sentences, from the original text without altering their content. This approach leverages statistical, linguistic, and semantic features to identify the most informative parts. Commonly used features include keyword frequency, sentence position, length, term centrality, and textual cohesion [Nenkova and McKeown, 2011].

Approaches can be broadly classified into unsupervised and supervised methods.

Unsupervised methods rely on statistical and graph-based techniques to estimate sentence importance without labeled data. Prominent examples include TextRank [Mihalcea and Tarau, 2004], which applies the PageRank algorithm to sentence graphs, and LexRank [Erkan and Radev, 2004], which leverages sentence similarity based on cosine similarity and centrality measures. Other techniques include Latent Semantic Analysis (LSA) [Gong and Liu, 2001] and Non-negative Matrix Factorization (NMF) [Wang *et al.*, 2008].

Supervised approaches treat summarization as a sentence classification or ranking problem, requiring annotated datasets. The pioneering work by [Kupiec *et al.*, 1995] applied a Naive Bayes classifier using surface-level features. Subsequent advances introduced more sophisticated models such as Conditional Random Fields (CRFs) [Galley, 2006] and neural architectures. Notably, Kågebäck et al. [Kågebäck *et al.*, 2014] proposed recursive autoencoders for learning sentence representations.

With the advent of deep learning and Transformers, models like BERTSUM [Liu and Lapata, 2019] and Siamese-BERT [Zhong *et al.*, 2020] have significantly improved the ability to capture semantic similarity and contextual dependencies. These models fine-tune pretrained language models to encode sentence-level relationships, enabling more coherent and semantically accurate extractive summaries.

Despite its simplicity and interpretability, extractive summarization can suffer from redundancy and lack of coherence between extracted sentences—limitations that abstractive methods aim to overcome.

2.3 Abstractive Summarization

Unlike extractive summarization, abstractive summarization generates new sentences that capture the essential meaning of the source text. This approach allows for more fluent, coherent, and concise summaries, often resembling how humans summarize information. However, it presents significant challenges due to the need for deep language understanding and natural language generation capabilities [Akashvarma *et al.*, 2024; Widyassari *et al.*, 2022].

Abstractive summarization techniques have evolved considerably over the years and can be broadly categorized into three main approaches: structure-based, semantic-based, and deep learning-based methods.

2.3.1 Structure-Based Approaches

Structure-based methods rely on analyzing the discourse and organizational structure of the input text to guide summary generation. These techniques often utilize tools such as discourse trees, topic segmentation, and graph-based representations to identify and preserve the logical flow of information [Gupta and Gupta, 2019; Barzilay and Lapata, 2008].

For example, discourse-based summarization leverages Rhetorical Structure Theory (RST) to model the relationships between text segments, enabling the generation of coherent summaries Marcu [2000]. Similarly, graph-based models construct semantic graphs where nodes represent sentences or concepts, and edges denote relationships, guiding content selection based on centrality or importance measures Ghalandari *et al.* [2020].

2.3.2 Semantic-Based Approaches

Semantic-based approaches focus on capturing the meaning and relationships within the text to ensure that the generated summary retains the intended semantics. These methods often employ semantic role labeling (SRL), Abstract Meaning Representation (AMR), and knowledge graphs to model deeper linguistic features [Banarescu *et al.*, 2013; Liu and Lapata, 2019].

For instance, Liu and Lapata [2019] propose a semantic graph-based approach that integrates AMR parsing with neural models to improve the semantic fidelity of summaries. Other works have explored the use of ontology-based frameworks and semantic embeddings to enhance the abstraction process Zhao *et al.* [2021]. Despite their potential, semantic-based methods are often combined with deep learning models to overcome limitations related to scalability and fluency.

2.3.3 Deep Learning for Summarization

Deep learning has been the driving force behind recent advances in abstractive summarization. Early models employed sequence-to-sequence architectures based on recurrent neural networks (RNNs) and long short-term memory (LSTM) networks to generate summaries [Nallapati *et al.*, 2016]. Subsequently, convolutional neural networks (CNNs) were explored for their ability to capture local patterns in text [Gehring *et al.*, 2017].

The introduction of the Transformer architecture [Vaswani *et al.*, 2017] marked a paradigm shift, enabling models to capture long-range dependencies more efficiently through self-attention mechanisms. Transformer-based models such as BART [Lewis *et al.*, 2020], PEGASUS [Zhang *et al.*, 2020], and T5 [Raffel *et al.*, 2020] have set new benchmarks in abstractive summarization by pretraining on large-scale corpora with generative objectives.

More recently, Large Language Models (LLMs) like GPT-3, GPT-4, PaLM, and Llama have demonstrated remarkable performance in zero-shot and few-shot summarization tasks, leveraging their vast pretraining to generalize across domains [Akashvarma *et al.*, 2024]. These models integrate knowledge, reasoning, and generation capabilities, reducing the need for task-specific supervised fine-tuning.

Despite their success, deep learning-based approaches face challenges related to faithfulness (avoiding hallucinations), factual consistency, and computational cost, which are active areas of research in the summarization community [Maynez *et al.*, 2020; Widyassari *et al.*, 2022].

2.4 Deep Learning Models for Summarization

Among the most relevant models for text summarization, the following stand out:

2.4.1 PEGASUS

The PEGASUS model [Zhang *et al.*, 2020] employs specialized pre-training for summarization tasks by masking entire sentences, allowing the model to learn how to reconstruct them. This enhances fine-tuning efficiency and achieves strong performance even with limited training data.

2.4.2 Models for the Portuguese Language

Notable summarization models tailored for Portuguese include:

- **PTT5-base-summ-xlsum** and **PTT5-base-summ-temario**: Based on the T5 architecture, fine-tuned for Portuguese summarization using datasets such as XL-Sum and TeMário [Paiola *et al.*, 2022].
- **ChatBode** and **GemBode**: Models adapted for Portuguese from InternLM2 and Gemma-7b, respectively, using QLoRA for refinement [Garcia *et al.*, 2024b].
- **Gemma 2-9b** and **Llama-3.1-8b-Instruct**: Large-scale models optimized for text understanding and generation, widely used for summarization and question-answering tasks [Team *et al.*, 2024].

2.5 Evaluation Challenges in Abstractive Summarization

Evaluating abstractive summaries remains a significant challenge in natural language processing. Traditional metrics such as ROUGE, BLEU, and METEOR rely on n-gram overlap between the generated summary and one or more reference summaries [Lin, 2004]. While effective for extractive summarization or tasks with high lexical similarity, these metrics often fail to capture the quality of abstractive summaries, which may involve paraphrasing, reordering, or using semantically equivalent expressions.

As a result, summaries that are semantically correct but lexically different from the reference may be penalized unfairly. This limitation has motivated the development of semantic-based evaluation metrics. Among the most widely adopted is BERTScore [Zhang *et al.*, 2019], which computes similarity based on contextual embeddings from pretrained language models, offering a better approximation of semantic similarity.

More recent efforts include metrics such as BLEURT [Sellam *et al.*, 2020], which combines pretrained representations with task-specific fine-tuning to better correlate with human judgments, and MoverScore [Zhao *et al.*, 2019], which measures the distance between word embeddings considering their distribution in the text.

In addition to automated metrics, human evaluation remains the gold standard for assessing abstractive summarization. Studies typically assess summaries along dimensions such as:

- **Faithfulness**: The degree to which the summary accurately reflects the information in the source text, without introducing hallucinated content [Maynez *et al.*, 2020].
- **Coherence**: Whether the summary is logically and grammatically well-formed.
- **Fluency**: The readability and linguistic naturalness of the summary.
- **Coverage**: Whether the summary captures the most relevant and important information from the source.

Despite advances, no single metric fully captures all aspects of summary quality, especially for abstractive models. Recent works suggest combining automatic metrics with targeted human evaluations to achieve a more reliable and comprehensive assessment [Fabbri *et al.*, 2021].

Given these challenges, the choice of evaluation metrics significantly influences the conclusions drawn about model performance. This has led to an increased emphasis on developing metrics that align better with human judgment, particularly in the era of large language models where summaries often diverge lexically while preserving meaning.

3 Methodology

This study evaluates the performance of various LLMs in abstract text summarization. The following sections outline the methodological approach, emphasizing result quality and computational efficiency.

3.1 Experimental Dataset

The experiment utilizes the RecognaSumm dataset [Paiola *et al*., 2024], comprising 135,272 samples split into training $(81, 163)$, validation $(27, 054)$, and test $(27, 055)$ sets. This is a newly developed and comprehensive dataset specifically designed for the task of automatic text summarization in Portuguese. It stands out due to its diverse origins, consisting of news articles collected from various sources, including news agencies and online portals. The dataset was built using web scraping techniques and meticulous curation, resulting in a rich and representative collection of documents covering various topics and journalistic styles. The creation of RecognaSumm aims to address a significant gap in summarization research for the Portuguese language, providing a training and evaluation resource that can be utilized for the development and enhancement of automated summarization models.

For evaluation, 30% of the test set $(8, 116$ samples) was randomly selected to balance computational efficiency and representativeness. This subset preserves topic diversity and textual variability, ensuring a robust assessment of the LLMs' generalization capabilities. The dataset includes texts of varying lengths and complexities, enabling a comprehensive evaluation of model robustness across different scenarios.

3.2 Prompt Engineering

Prompt engineering is a crucial step, as it defines how the language model processes and responds to the summarization task. In this study, three prompt variations will be tested to determine which yields the best performance based on BLEU and ROUGE metrics. The prompt engineering techniques that will be applied to optimize the interaction between models and the task are Zero-Shot, One-Shot and Few-Shot Prompting.

For One-Shot and Few-Shot Prompting, one and four examples will be drawn from the training data, respectively, and 100 examples from the validation set will be tested against three candidate models.

The prompts with the examples are showned in Appendix A.

To assess the quality and precision of summarization, BLEU and ROUGE scores will be computed between the generated summaries and the reference summaries from the 100 examples. The most effective prompt will then be selected for further experimentation across all language models.

3.2.1 Zero-Shot Prompting

This method requires the model to generate summaries solely based on a direct instruction, without any reference examples. The prompt simply states, *"Summarize the following text,"* relying on the model's general knowledge to determine the appropriate structure and content. While this approach is effective for straightforward tasks, it may struggle with maintaining consistency in summarization style or capturing subtle nuances, as no explicit guidance is provided [Brown *et al*., 2020].

3.2.2 One-Shot Prompting

In one-shot prompting, the model is given an instruction to generate a summary, accompanied by a single example demonstrating the expected format. This example provides a structural reference, helping the model align its output with the desired level of detail. For instance, the prompt might include the instruction *"Summarize the following text,"* followed by a sample summary of a similar passage. By referencing this example, the model can better identify key information and produce summaries with improved accuracy [Brown *et al*., 2020].

3.2.3 Few-Shot Prompting

The few-shot prompting technique involves presenting task instructions alongside multiple input-output examples. By doing so, the model gains a clearer understanding of the expected response format and stylistic requirements, improving the accuracy of its outputs [Liu *et al*., 2021]. Even with a small set of examples, this method can effectively guide the model's behavior toward the desired outcome.

3.3 Selection of Language Models

The study compares the performance of six language models. The selected LLMs were chosen based on their summarization capabilities and availability:

- Chatbode (Recogna): Developed by Recogna, this model is optimized for dialogue-based interactions and is capable of generating clear and concise responses. Its architecture is designed for natural language processing tasks involving textual understanding and generation based on specific instructions, making it well-suited for summarization tasks [Garcia *et al*., 2024b];
- GemBode (Recogna): Another Recogna model, GemBode is a larger and more robust version of Chatbode, focusing on tasks requiring deeper semantic understanding. Designed for complex contexts, GemBode is expected to perform well in summarization tasks that require precision and information integrity [Garcia *et al*., 2024a];
- ptt5-base-summ-xlsum (Recogna): A variant of the T5 (Text-to-Text Transfer Transformer) model, specifically fine-tuned for summarizing long texts such as reports or articles. The ptt5-base-summ-xlsum has been trained on large volumes of summarization data, making it particularly effective at generating detailed summaries without compromising textual cohesion [Paiola *et al*., 2022];
- ptt5-base-summ-temario (Recogna): Another T5-based variant, this model specializes in thematic summarization. It is trained to generate summaries that highlight central themes, which can be particularly useful when summaries need to focus on key topics within extensive texts [Paiola *et al*., 2022];

- Gemma-2-9B (Google): A large-scale model with 9 billion parameters, the Gemma-2 is designed to generate text with high coherence and contextual relevance. Due to its size and robust processing capabilities, the Gemma-2-9B is ideal for text generation tasks requiring deep semantic understanding, handling complex contexts, and producing high-quality summaries [Team *et al.*, 2024];
- Llama-3.1-8B-Instruct (Meta): Part of the LLaMA (Large Language Model Meta AI) series developed by Meta AI, this model is fine-tuned to follow instructions with high precision, making it highly effective in supervised tasks such as summarization. The Llama-3.1-8B-Instruct model, with 8 billion parameters, is known for its ability to follow detailed prompts while maintaining a good balance between length and coherence in generated texts [Dubey *et al.*, 2024].

These models will be evaluated using the selected prompts, and their performance will be compared based on BLEU and ROUGE scores to determine the most effective model for abstractive summarization.

3.4 Sample Size Assessment

A preliminary analysis of the dataset will be conducted to examine the number of characters and tokens in each text sample. This step ensures that all inputs remain within the token limitations of the LLMs. If some samples exceed these constraints, preprocessing techniques will be applied, such as strategic truncation—retaining essential content—or breaking the text into smaller segments for processing.

3.5 Model Execution and Performance Metrics

Once the best prompt has been selected and the parameters have been defined, the LLMs will be tested using $8,116$ samples from the evaluation set. The following metrics will be applied to assess performance:

- BLEU: Evaluates precision by measuring the overlap of n-grams between generated outputs and reference texts;
- ROUGE: Focuses on recall, determining phrase overlap between model-generated summaries and human-written references;
- Compression Rate: Calculates the reduction in text length when comparing generated summaries to the original passages;
- Inference Time: Measures the average processing time per sample, including variance, to analyze computational efficiency.

3.6 Quantization Tests

The impact of quantization at different precision levels (4-bit and 8-bit) will be examined. Quantization reduces model size and enhances computational efficiency while maintaining accuracy. Comparisons will be made between quantized and standard versions to analyze trade-offs in performance.

3.7 Expected Results

The findings will be analyzed from two perspectives:

- Summarization Quality: BLEU and ROUGE scores will be compared across models, supplemented by human evaluation of selected summaries;
- Computational Efficiency: Inference time and quantization effects will be assessed to determine the best balance between quality and efficiency.

4 Experiments

This section will cover the experiments of this study.

4.1 Number of tokens in the input texts

Initially, the number of tokens was calculated for each of the 8,116 input texts in the dataset using the tokenizers employed in the experiment. This allows for determining an appropriate number of tokens for the tokenization process, as well as assessing whether text truncation will be necessary. Table 1 presents the aggregated statistics of the token count distributions for the input texts for each tokenizer.

It can be observed that the number of tokens varies across tokenizers. The averages range between 600 and 900 tokens, while the maximum values exceed $10,000$ tokens.

To preserve essential information from long texts while ensuring good computational efficiency, a limit of $1,024$ tokens was chosen for all tokenizers to maintain consistency. This value ensures that inputs remain within well-defined limits, preventing excessive truncation or unnecessary computational overhead.

4.2 Number of tokens in the reference summaries

To determine the number of tokens generated by the models, an analysis of the number of tokens in the reference summaries was conducted. The results of the aggregated statistics of the distributions per tokenizer are presented in Table 2.

As can be observed, the reference summaries contain up to 203 tokens. Thus, it is possible to select a token count of 256 tokens or more in the generation process, depending on the desired level of summarization, to standardize the outputs across all models. In this experiment, 1024 tokens were used, a value chosen to avoid restricting the generated summaries to only 256 tokens, allowing the models to decide when to stop.

4.3 Tokenizers and Models Parameters

To ensure a fair comparison, the same parameters were used for both the tokenizers and the generation of new tokens. The code was implemented in the Python programming language, using the Transformers, PyTorch, and HuggingFace libraries.

Table 1. Aggregated statistics of the token count distributions for the input texts.

Statistic	llama	gemma	chatbode	gembode	ptt5_xlsum	ptt5_temario
Mean	761.67	641.03	911.06	641.03	603.43	603.43
SD	772.27	654.46	918.52	654.46	619.11	619.11
Min	2.00	2.00	2.00	2.00	3.00	3.00
25%	301.00	253.75	360.00	253.75	238.00	238.00
50%	527.00	443.00	631.50	443.00	416.00	416.00
75%	936.25	784.00	1117.25	784.00	739.00	739.00
Max	10534.00	9130.00	11722.00	9130.00	8699.00	8699.00

Table 2. Aggregated statistics of the token count distributions for the reference summaries

Statistic	llama	gemma	chatbode	gembode	ptt5_xlsum	ptt5_temario
Mean	65.09	54.44	77.03	54.44	50.70	50.70
SD	16.96	14.67	20.22	14.67	13.43	13.43
Min	15.00	12.00	16.00	12.00	12.00	12.00
25%	53.00	44.00	62.00	44.00	41.00	41.00
50%	63.00	53.00	75.00	53.00	49.00	49.00
75%	76.00	63.00	89.00	63.00	58.00	58.00
Max	162.00	136.00	203.00	136.00	138.00	138.00

4.3.1 Tokenizer Parameters

- **max_length**: Limits the maximum number of tokens generated by the tokenizer.
  - Value (1024): Texts exceeding 1024 tokens will be truncated or adjusted.
- **truncation**: Indicates whether texts longer than the limit (max_length) should be truncated.
  - Value (True): Ensures that longer texts are cut to fit within the maximum length.
- **return_tensors**: Specifies the format of the tensor returned by the tokenizer.
  - Value ('pt'): Returns tensors in the PyTorch format, used as input for the models.

4.3.2 Model Parameters

- **do_sample**: Defines whether generation should use stochastic sampling or the highest probability token.
  - Value (True): Enables sampling, allowing greater variability in responses.
- **early_stopping**: Determines whether the model should stop generating tokens when a stopping condition is met.
  - Value (False): The model will continue until it reaches the limit defined by max_new_tokens.
- **min_length**: Defines the minimum number of tokens the model should generate.
  - Value (32): Ensures that summaries are not excessively short.
- **max_new_tokens**: Limits the maximum number of new tokens generated by the model.
  - Value (1024): Allows the model to generate up to 1024 additional tokens.
- **num_beams**: Configures the number of search paths in the beam search algorithm.
  - Value (1): With num_beams=1, beam search is disabled, and the model relies solely on sampling.
- **num_return_sequences**: Specifies the number of distinct sequences generated for each input.
  - Value (1): Only one output sequence is generated per text.
- **temperature**: Controls the randomness of generation by adjusting probability smoothing.
  - Value (0.3): Low temperature, favoring more predictable choices.
- **top_p**: Uses nucleus sampling to limit generation to words whose cumulative probability sum reaches top_p.
  - Value (0.8): Restricts generation to the most probable words until 80% of the cumulative probability is reached.
- **top_k**: Limits the choice of tokens to the k most probable ones.
  - Value (80): Ensures that the model selects only from the 80 most probable tokens at each step.

These parameters were chosen to balance control and creativity in generation. The use of sampling with top_p, top_k, and temperature promotes diversity, while max_new_tokens and min_length ensure that the results remain within expected limits. The tokenizer parameters ensure compatibility and uniformity in input texts, standardized with up to 1024 tokens for consistency in the experiment.

4.4 Prompt Development

Initially, an experiment was conducted to determine the best prompt to be used with the models. This test was performed using 100 samples from the dataset in question, and 3 prompts were tested for 4 out of the 6 models (the **ptt5** models were fine-tuned without a specific prompt for summarization, where given an input text without any additional information or instructions, the models summarize the text), as mentioned in section 3.2. The results are shown in Table 3.

The BLEU and ROUGE metrics were predominantly better for the zero-shot prompt for most models, except for the **ChatBode** model, which achieved a better BLEU score with the zero-shot prompt, but performed better in ROUGE with the one-shot prompt, as shown in Table 3.

Language models are trained at a large scale and have strong generalization capabilities. They learn language patterns from vast amounts of data, enabling them to generate high-quality summaries without additional examples, as in the case of zero-shot. Another relevant factor is that in zero-shot, there is no interference from specific examples, avoiding contextual noise.

The BLEU and ROUGE metrics assess lexical and structural similarity between the model's output and the reference texts. Models that capture summarization patterns well can achieve good metrics even without specific examples.

The superior performance of the zero-shot prompt suggests that the models have a strong ability to generalize for the summarization task. The isolated advantage of **ChatBode** in the one-shot prompt for ROUGE may indicate that its architecture or training benefits from more additional context to identify key terms.

Based on this experiment, it can be concluded that the zero-shot prompt is more suitable for text summarization in this specific context. Therefore, the experiment will proceed with this prompt.

4.5 Summarization with 4-bits quantization

At this stage, the six models were evaluated on $8,116$ samples from the test set. The same subset of $8,116$ samples was consistently used to ensure fairness. Metrics such as BLEU, ROUGE, Compression Rate, and Inference Time were collected. The results of the 4-bit experiment are presented in the following sections.

4.5.1 BLEU and ROUGE

In Figure 1, the graph with the metric values by model can be observed.

The results show that **Chatbode** achieved the best performance in almost all metrics (BLEU, ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum), indicating that this model successfully captured both the lexicality and cohesion of the text.

Llama showed the worst performance across all metrics, suggesting that this model struggled to generate relevant and coherent summaries.

Gembode and **Gemma** performed very similarly, with **Gembode** having a slight advantage in BLEU, while **Gemma** performed slightly better in ROUGE-2. This is expected since the **Gembode** model is a fine-tuned version of the **Gemma** model for the summarization task.

4.5.2 Compression Rate

The Compression Rate (CR) [Liu *et al.*, 2022] is defined as the length of the generated summary divided by the length of the text to be summarized. A CR smaller than 1 indicates that the generated summary is shorter than the original text, while a CR greater than 1 indicates that the generated summary may be excessively long or even incorrect. In Table 5, the aggregated statistics of these distributions for each of the models are presented. The values shown include the mean, standard deviation, minimum and maximum values, and quartiles (25%, 50%, and 75%).

To avoid distortion effects in the aggregated statistics of the distributions due to extreme values in the maxima, for CR values above 1, the values were adjusted to 1. The result of the new aggregated statistics can be seen in Table 6, along with the distributions in Figure 2.

It can be observed that after the adjustment, the models largely maintained their aggregated statistics, except for **Llama**, which had all its quartiles influenced by the adjustment. The differences in the means are also noticeable, as this is a statistic strongly influenced by outliers.

The models **Chatbode** and **Gembode** show similar means (0.387 and 0.408) and controlled standard deviation, making them ideal for tasks that require compact summarization with limited variation. The **ptt5_temario** model is slightly more expansive (mean of 0.432), but with high consistency.

On the other hand, the **ptt5_xlsum** model has the lowest mean (0.204), reflecting its preference for extremely short summaries. It may be useful in contexts that require significant content reduction. This is again explained by how **ptt5_xlsum** was designed, with fine-tuning performed for the summarization task on a dataset with more compact reference summaries.

The **Llama** model, even after adjustment, still shows unsatisfactory summarization results, with a high mean (0.847) and values frequently hitting the adjusted limit. The **Gemma** model, despite having an acceptable mean (0.458), has a high standard deviation (0.323) and considerable variability, making it less reliable.

4.5.3 Inference Time

The Inference Time (TI) is given by the total time for generating the summary, measured in seconds. This is a relevant metric regarding the computational efficiency of the models. In figure 3, the TI distributions are shown, and in table 7, the aggregated statistics of these distributions by models are presented.

The **ptt5_xlsum** model is the best option for scenarios requiring fast inference. The **ptt5_temario** model combines speed with good consistency, making it suitable for balancing efficiency and quality.

The **Chatbode** model is a good compromise between inference time and variability, while the **Gembode** model showed acceptable times in most inferences but has significant outliers.

On the other hand, the **Llama** and **Gemma** models are slow, have high variability, and may be unsuitable for time-sensitive tasks.

Table 3. BLEU and ROUGE metrics for the prompt experiment (values shown as percentages).

Model	Prompt	BLEU (%)	ROUGE-1 (%)	ROUGE-2 (%)	ROUGE-L (%)	ROUGE-Lsum (%)
Gemma	Zero-shot	0.93	23.60	6.32	15.45	15.51
Gemma	One-shot	0.24	6.66	1.43	4.87	5.28
Gemma	Few-shot	0.00	0.03	0.00	0.03	0.03
Gembode	Zero-shot	0.99	23.95	6.62	15.57	15.46
Gembode	One-shot	0.95	10.97	2.85	8.13	8.45
Gembode	Few-shot	0.00	0.03	0.00	0.03	0.03
Chatbode	Zero-shot	0.86	23.82	3.65	14.95	14.97
Chatbode	One-shot	0.64	26.49	4.44	17.66	17.65
Chatbode	Few-shot	0.00	25.27	4.77	17.29	17.29
Llama	Zero-shot	0.20	7.69	1.92	5.40	6.26
Llama	One-shot	0.12	6.28	1.03	4.59	5.27
Llama	Few-shot	0.00	0.03	0.00	0.03	0.03

Table 4. Metrics for 4-bit execution (values shown as percentages).

Model	BLEU (%)	ROUGE-1 (%)	ROUGE-2 (%)	ROUGE-L (%)	ROUGE-Lsum (%)
chatbode	5.71	30.04	13.41	20.89	20.90
gembode	4.21	23.90	10.47	16.83	16.85
ptt5_temario	3.88	24.99	10.68	17.47	17.48
ptt5_xlsum	3.35	28.29	7.74	18.59	18.59
llama	0.88	7.44	2.89	5.58	5.70
gemma	3.49	23.69	10.86	16.82	16.81

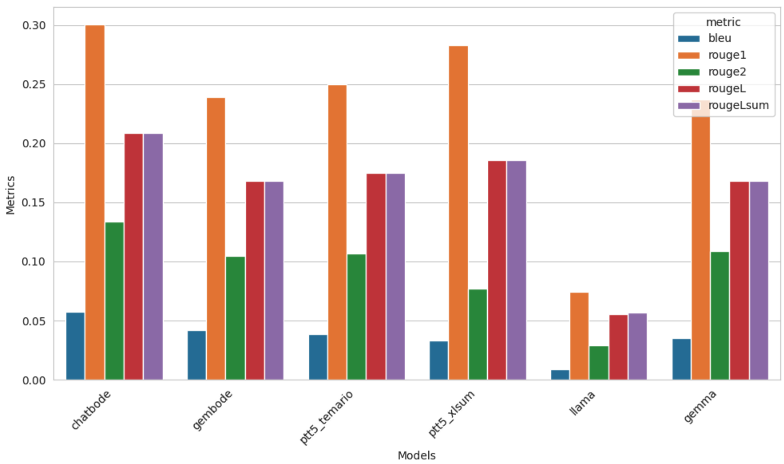

Figure 1. Comparison of metrics in 4-bits execution

Table 5. Compression Rate of generated summaries in 4-bits quantization (without correction), values shown with two decimal places.

Model	Chatbode	Gembode	ptt5_temario	ptt5_xlsum	Llama	Gemma
Mean	0.72	0.56	0.76	0.57	6.36	4.46
SD	8.44	5.18	7.65	8.57	95.79	108.39
Min	0.00	0.00	0.01	0.01	0.00	0.00
25%	0.21	0.19	0.31	0.08	1.01	0.20
50%	0.34	0.36	0.41	0.14	1.87	0.40
75%	0.52	0.60	0.53	0.25	3.32	0.71
Max	376.00	432.00	484.00	278.00	3632.00	4545.00

Table 6. Compression Rate for generated summaries in 4-bits quantization (with correction), values shown with two decimal places.

Model	Chatbode	Gembode	ptt5_temario	ptt5_xlsum	Llama	Gemma
Mean	0.39	0.41	0.43	0.20	0.85	0.46
SD	0.23	0.29	0.19	0.20	0.31	0.32
Min	0.00	0.00	0.01	0.01	0.00	0.00
25%	0.21	0.19	0.31	0.08	1.00	0.20
50%	0.34	0.36	0.41	0.14	1.00	0.40
75%	0.52	0.60	0.53	0.25	1.00	0.71
Max	1.00	1.00	1.00	1.00	1.00	1.00

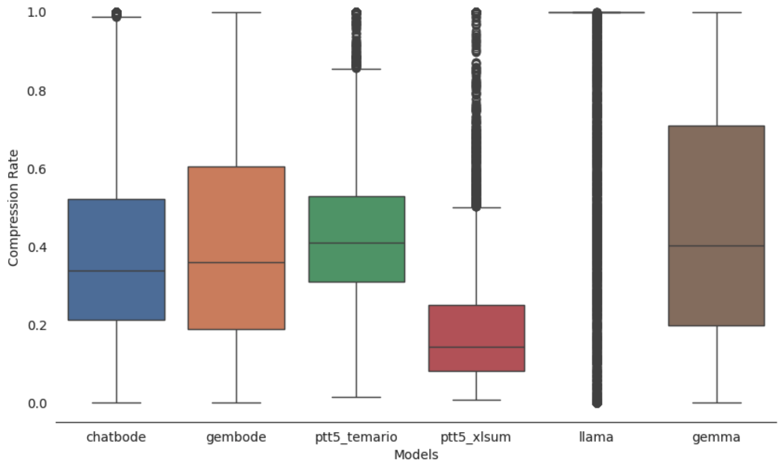

Figure 2. Distribution of CRs in 4-bits quantization

Table 7. Aggregated statistics of the Inference Time distributions for the models with 4 bits quantization, measured in seconds.

Statistic	Chatbode	Gembode	ptt5_temario	ptt5_xlsum	Llama	Gemma
Mean	17.13	29.80	12.75	3.29	66.35	44.83
SD	12.96	34.61	12.15	2.60	10.93	47.03
Min	0.14	0.30	1.40	2.04	2.83	0.50
25%	10.72	10.19	5.12	2.74	59.80	14.32
50%	14.40	16.68	9.56	3.11	67.57	21.93
75%	18.79	29.55	16.37	3.56	74.57	51.02
Max	121.12	148.13	80.92	79.93	86.67	153.71

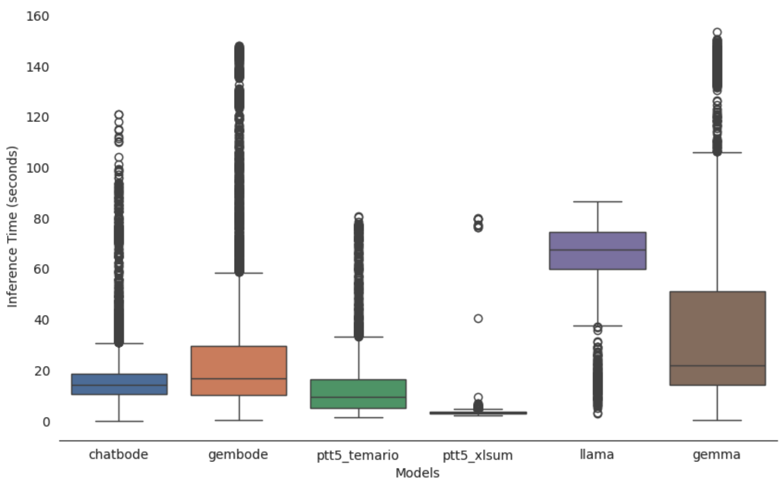

Figure 3. IT Distribution in 4-bits quantization

4.6 Summarization with 8-bits quantization

In the previous section, we observed the performance of the models when quantized to 4 bits. In this section, the models were quantized to 8 bits, and the experiment was repeated on the $8,116$ samples of the test set. The BLEU, ROUGE, Compression Rate, and Inference Time metrics were collected. The results of the 8 bits experiment are presented in the following sections.

4.6.1 BLEU and ROUGE

The metrics can be compared graphically in Figure 4.

The results indicate that **Chatbode** had the best overall performance, achieving the highest scores in all metrics, especially in ROUGE-1, ROUGE-2, and ROUGE-L, which suggests that the model was effective in both content coverage and capturing the structure of the text. **Gemma** also achieved good scores, especially when compared to the T5-based models like **ptt5_xlsum** and **ptt5_temario**, which showed more modest results, particularly in terms of ROUGE-2 and ROUGE-L. **Llama** was the model that obtained the worst results in the BLEU and ROUGE metrics, suggesting that, in this specific experiment, it did not perform well for the summarization task.

4.6.2 Compression Rate

In figure 5 and in tables 9 and 10, the distributions and aggregated statistics of the Compression Rates of the models are shown, this time for the 8 bits quantization. As in the 4 bits experiment, the results will be presented with and without correction (for TC values above 1, they were adjusted to 1). The values presented are mean, standard deviation, minimum and maximum values, and quartiles (25%, 50%, and 75%). For figure 5, the distributions shown are with correction.

Llama stands out for generating summaries that, in most cases, maintain the original size of the text (compression rate of 1.0), suggesting a non-compression approach or more informative summaries, possibly sacrificing conciseness.

The models **ptt5_xlsum** and **ptt5_temario** demonstrated more aggressive compression, with lower compression rates, which may result in shorter and possibly more synthetic summaries.

Chatbode, **Gembode**, and **Gemma** show a wide variation in compression rates, indicating that they can generate summaries that are either very short or longer, providing a balance between conciseness and content.

The **ptt5_xlsum** model stands out for its extremely low compression rate, indicating that it may be more suitable for tasks where only the most relevant content is desired, although there is a risk of losing significant information.

The presence of low minimum values (close to zero) in many of the models suggests that in some cases the summaries may be excessively short, which could harm the quality of the generated content.

4.6.3 Inference Time

The figure 6 and table 11 display the distributions and aggregated statistics of the models' Inference Time.

The models exhibit significant variation in inference time, with some, such as **ptt5_xlsum**, offering very fast results, while others, like **Llama**, have high inference times.

Ptt5_xlsum stands out as the fastest and most consistent model, making it suitable for real-time or large-scale applications.

Models like **Chatbode** and **Gembode** show higher times with greater variation, which could impact consistency in applications requiring quick responses.

Llama and **Gemma** have longer inference times, which could be a limiting factor in large-scale processing or low-latency contexts.

Ptt5_temario is an intermediate model, performing well in terms of inference time, suitable for tasks that require a balance between speed and generation quality.

4.7 Comparison between quantizations

Quantization is a technique that reduces the size of models by representing weights and activations with lower precision. 8 bit quantization offers good efficiency with minimal quality loss, as well as greater compatibility with hardware and optimized frameworks, making it ideal for applications that prioritize stable performance. However, the size reduction is less significant compared to 4 bit quantization.

On the other hand, 4 bit quantization drastically reduces the model size, making it ideal for devices with limited memory, as well as speeding up execution and consuming less power. However, there may be more quality degradation and instabilities, with less support in some frameworks and hardware. The choice between the two depends on the desired balance between quality, compression, and available resources.

4.7.1 Comparison in terms of BLEU and ROUGE

The 4 bit quantization maintained competitive performance in the models **ptt5_xlsum**, **ptt5_temario**, and **Chatbode**, with BLEU and ROUGE close to or even better than the 8 bit version. However, models such as **Gembode**, **Llama**, and **Gemma** showed more noticeable drops, with differences such as ROUGE1 dropping from 0.260 to 0.239 in **Gembode**. This suggests that the effectiveness of quantization depends on the model, making it a viable solution to save resources in cases where slightly reduced performance is acceptable.

Figure 7 shows the BLEU, ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Sum metrics comparing the 4 and 8 bit quantizations.

4.7.2 Comparison in terms of Compression Rate

The analysis of the compression rates shows that the 4 bit quantization is generally competitive, with averages close to or even higher than those of the 8 bit version in some cases, such as in the models **ptt5_xlsum** and **Gemma**. Although the standard deviation is slightly higher in 4 bits in some models, indicating greater variability, the compression achieved is satisfactory for applications that prioritize computational efficiency. Models like **Llama**, with inherently high compression, were less affected by the reduction in precision,

Table 8. Execution metrics of the models with 8-bit quantization (values shown as percentages).

Model	BLEU (%)	ROUGE-1 (%)	ROUGE-2 (%)	ROUGE-L (%)	ROUGE-Lsum (%)
Chatbode	5.40	29.23	13.09	20.29	20.30
Gembode	3.40	26.00	11.16	18.39	18.37
ptt5_temario	3.35	23.71	10.21	16.84	16.83
ptt5_xlsum	3.62	28.10	7.77	18.81	18.82
Llama	0.93	7.61	3.00	6.14	5.95
Gemma	3.76	27.17	12.29	19.21	19.20

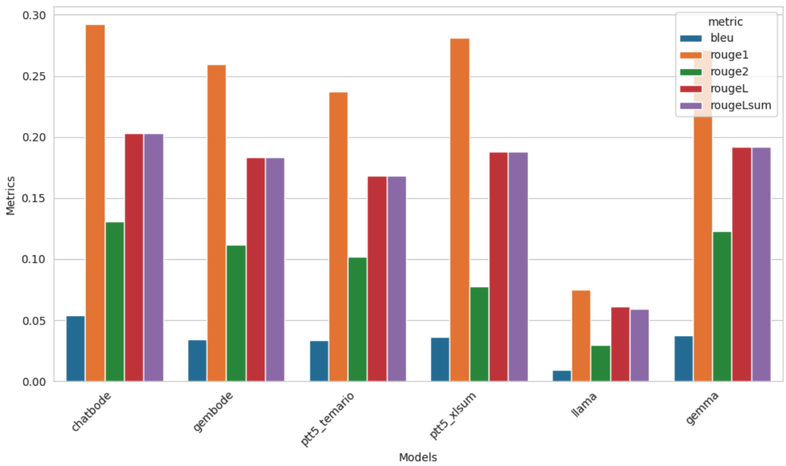

Figure 4. Comparison of metrics with 8-bits quantization

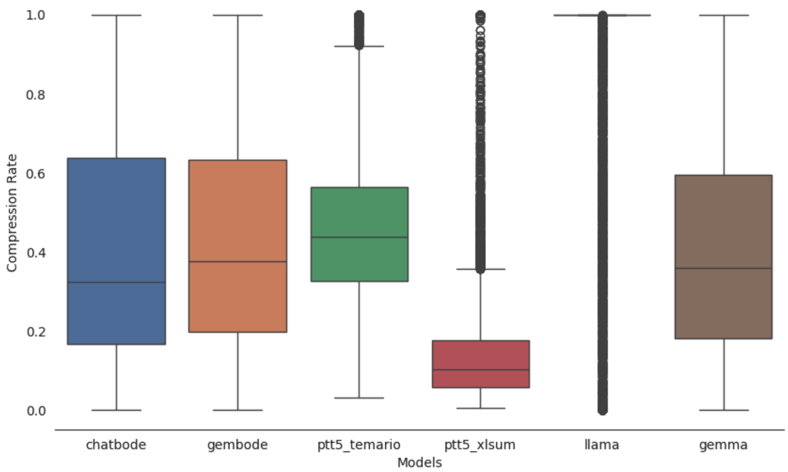

Figure 5. CT Distributions in 8-bits quantization

Table 9. Compression Rate for generated summaries in 8-bit quantization (without correction), values shown with two decimal places.

Statistics	Chatbode	Gembode	ptt5_temario	ptt5_xlsum	Llama	Gemma
Mean	1.53	0.63	0.84	0.40	9.03	7.19
SD	46.77	7.01	8.37	6.07	137.06	169.26
Min	0.00	0.00	0.03	0.00	0.00	0.00
25%	0.23	0.20	0.33	0.06	1.11	0.18
50%	0.36	0.37	0.44	0.10	1.97	0.36
75%	0.55	0.63	0.56	0.18	3.50	0.59
Max	3163.00	483.62	484.00	207.00	3835.00	5488.00

Table 10. Compression Rate for generated summaries in 8-bit quantization (with correction), values shown with two decimal places.

Statistics	Chatbode	Gembode	ptt5_temario	ptt5_xlsum	Llama	Gemma
Mean	0.41	0.42	0.46	0.15	0.85	0.40
SD	0.24	0.30	0.20	0.16	0.31	0.30
Min	0.00	0.00	0.03	0.00	0.00	0.00
25%	0.23	0.20	0.33	0.06	1.00	0.18
50%	0.36	0.37	0.44	0.10	1.00	0.36
75%	0.55	0.63	0.56	0.18	1.00	0.59
Max	1.00	1.00	1.00	1.00	1.00	1.00

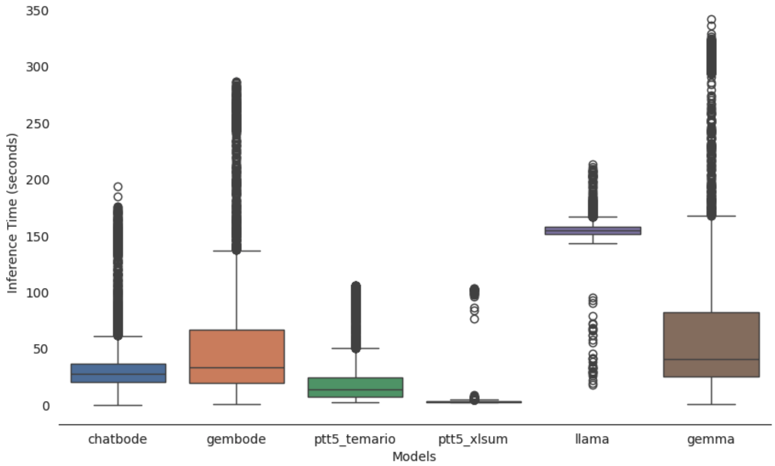

Figure 6. IT Distribution in 8-bits quantization

Table 11. Aggregated statistics of the models' Inference Time distributions for 8 bits quantization, measured in seconds.

Statistics	Chatbode	Gembode	ptt5_temario	ptt5_xlsum	Llama	Gemma
Mean	34.16	67.46	20.82	3.44	155.79	83.44
SD	27.75	79.37	22.25	5.78	11.01	99.31
Min	0.19	0.53	1.92	1.89	19.67	0.44
25%	20.22	19.37	7.47	2.55	151.29	25.10
50%	27.15	32.65	13.85	2.95	154.81	40.50
75%	36.66	66.34	24.50	3.49	158.31	81.45
Max	194.11	286.41	105.85	103.43	213.65	328.87

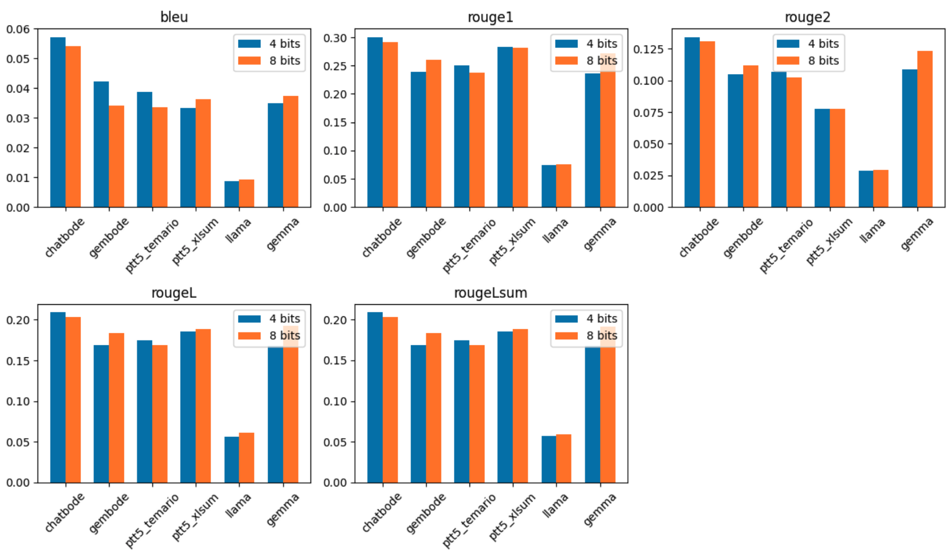

Figure 7. Metrics for models in 4 and 8 bits quantizations

while **ptt5_xlsum** and **Gemma** stood out in efficiency with the more aggressive quantization.

4.7.3 Comparison in terms of Inference Time

As expected, the 4 bit quantization resulted in lower inference times in all models compared to the 8 bit version, with significant average reductions, such as in **Chatbode** and **Llama**. Consistency in times (assessed by the standard deviation) also improved in 4 bits for most models, suggesting that 4 bit quantization not only accelerated inferences but also maintained stability. Lighter models like **ptt5_xlsum** showed less impact from quantization, while more complex models, like **Llama**, benefited significantly. Therefore, 4 bit quantization is a superior choice in scenarios that require fast inferences without compromising stability.

4.8 Discussion

Based on the results obtained, we can evaluate the performance of the models considering summarization quality metrics (BLEU, ROUGE), compression rates, and inference times, both for 4 bit and 8 bit quantizations.

4.8.1 Best Model

The **Chatbode** model showed the best balance between summarization quality, compression rate efficiency, and inference time.

- In terms of quality metrics, it consistently achieved the highest BLEU (0.057146 in 4 bits and 0.054023 in 8 bits) and ROUGE values across all variants, outperforming the other models.
- In terms of compression rate, it maintained competitive average values, around 0.387 in 4 bits and 0.409 in 8 bits, with good consistency.
- The inference time in 4 bits was significantly reduced (average of 17.13 seconds) with minimal loss in stability, while in 8 bits it was moderate (34.16 seconds).

These results make **Chatbode** the best choice for scenarios that require high-quality summarization with processing efficiency.

4.8.2 Worst Model

The **Llama** model was consistently the least efficient across all evaluated metrics:

- In terms of quality metrics, it showed the lowest values, with BLEU of 0.008768 in 4 bits and 0.009294 in 8 bits, and similar performance in ROUGE.
- Despite high compression rates (0.847 in 4 bits and 0.853 in 8 bits), the model exhibited low variability in the length of generated texts, which may indicate a lack of flexibility.
- The inference time was the highest among all models, with an average of 66.35 seconds in 4 bits and an impressive 155.79 seconds in 8 bits, making it impractical for applications that require fast responses.

The results indicate that the **Chatbode** model is the most balanced, showing an excellent trade-off between summarization quality, compression rate, and temporal efficiency. In contrast, the **Llama** model performed poorly across all dimensions, especially in more computationally demanding scenarios. For practical applications, models such as **Chatbode** and, secondarily, **Gemma** and **ptt5** models, are more suitable choices, with the recommendation to prioritize 4-bit quantization to maximize computational efficiency without significantly compromising quality.

4.8.3 Quality of Generated Summaries

Overall, the generated summaries reveal varying levels of abstraction, conciseness, and factual alignment with the reference, as it can be seen in Appendix B. Models like **ptt5_temario**, **llama**, and **gemma** produce outputs that closely match both the factual content and key numerical details from the source text, effectively summarizing the core information about Biden's approval of the debt ceiling increase and the $2.5 trillion figure. **chatbode** and **gembode** also capture the main facts and emphasize the prevention of an unprecedented default, reflecting a good understanding of details, but with slightly less numeric precision in some cases. **Ptt5_xlsum** generates a more verbose and repetitive summary, with redundant mentions of the government's announcement and missing specific figures, which diminishes its informativeness and conciseness. Compared to the concise reference summary, most model outputs add contextual details (e.g., avoiding default, procedural steps) that, while accurate, may exceed the desired brevity depending on the summarization task. These differences underscore how abstractive summarization models vary in selecting salient content, maintaining factual accuracy, and balancing brevity with informativeness.

4.8.4 Limitations on the use of the dataset

An important limitation of this study lies in the reliance on a single dataset, RecognaSumm, which may constrain the generalization of the results to other contexts or thematic domains. Although the corpus contains a significant number of samples, providing a robust basis for experimentation, it is possible that summarization models might exhibit different behavior when applied to texts from other areas, especially technical documents or specialized content. Nevertheless, the results obtained remain a valuable indication of the models' performance and provide relevant insights into the current capabilities and limitations of abstractive summarization approaches. Although we did not conduct human linguistic evaluations or usability testing, the automatic metrics and large-scale dataset employed provide a robust initial assessment of model performance, laying the groundwork for future human-centered studies.

5 Conclusion

The experiment conducted aimed to evaluate the performance of six LLMs in the task of abstractive text summarization, using three prompting strategies (zero-shot, one-shot, and few-shot) and evaluation metrics (BLEU, ROUGE, inference time, compression rate, and number of tokens). From the obtained results, it was possible to identify relevant patterns and highlight the main trade-offs between quality, efficiency, and model suitability for the task in question.

The results indicated that the zero-shot strategy showed consistent performance in the BLEU and ROUGE metrics for most models, suggesting that, in many situations, LLMs are capable of generalizing to summarization tasks without the need for additional examples. However, an exception was observed where the **Chatbode** model showed better ROUGE in the one-shot configuration, indicating that the inclusion of examples can be beneficial in specific contexts.

Regarding the BLEU and ROUGE metrics in the main experiment, it was found that the **Chatbode** model achieved the best results, while the **Llama** model had the worst results. **Chatbode** exhibited the best balance across all metrics, making it the most suitable model for more generic practical applications without significant bottlenecks. In contrast, the **Llama** model was found to be unfeasible for the text summarization task.

The analysis of the compression rate revealed that standardizing to 1024 tokens was a good decision, balancing the preservation of relevant content and the limitations of each model. Additionally, adjusting values above 1 demonstrated the impact of outliers on aggregated calculations and the importance of statistical controls.

Regarding inference time, significant differences between models were identified, with a focus on the efficiency of the **ptt5_xlsum** model and the higher slowness of the **Llama** model. This variation highlights the need to consider not only the quality of outputs but also the associated computational costs, especially in practical application scenarios.

The parameters used for tokenization and text generation also played a fundamental role in controlling the experimentation, allowing a balance between diversity and accuracy in the responses generated by the models. The temperature, top-k, and top-p settings proved to be suitable for avoiding repetitions and promoting creative outputs.

Finally, the experiment reinforces the importance of methodology in data science and artificial intelligence projects, as well as detailed analyses in the evaluation of LLMs for specific tasks. The results contribute not only to the field of text summarization but also to the practice of applying LLMs to real-world problems, providing valuable insights for future work in the area.

Declarations

Acknowledgements

The authors would like to thank ICMC/USP and UNESP for the support and resources provided during this research. The authors would like to thank FAPESP and CNPq for the funding of this paper. Finally, we acknowledge the contributions of the open-source community, whose tools and frameworks were fundamental to this study.

Authors' Contributions

Hugo Alexandre Padovani Guimarães de Camargo contributed to the conceptualization, data curation, formal analysis, investigation, methodology, project administration, software, resources, validation, visualization and writing of the original draft. Pedro Henrique Paiola, Gabriel Lino Garcia and João Paulo Papa performed supervision and writing - review & editing. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Funding

The authors are grateful to the São Paulo Research Foundation (FAPESP) grants 2013/07375-0, 2023/14427-8, and 2024/00789-8, and also to the Brazilian National Council for Scientific and Technological Development (CNPq) grants 308529/2021-9 and 400756/2024-2.

Availability of data and materials

The datasets (and/or softwares) generated and/or analysed during the current study will be made upon request.

References

Akashvarma, M. *et al*. (2024). A comprehensive review of large language models in abstractive summarization of news articles. In *Proceedings of the 2024 Asia Pacific Conference on Innovation in Technology (APCIT)*, pages 1–6. IEEE. DOI: 10.1109/apcit62007.2024.10673650.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Bulgaria. Association for Computational Linguistics. Available at:`https://aclanthology.org/W13-2322.pdf`.

Barzilay, R. and Lapata, M. (2008). Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34. DOI: 10.1162/coli.2008.34.1.1.

Basyal, L. and Sanghvi, M. (2023). Text summarization using large language models: A comparative study of mpt-7b-instruct, falcon-7b-instruct, and openai chat-gpt models. DOI: 10.48550/arXiv.2310.10449.

Brown, T. *et al*. (2020). Language models are few-shot learners. In *NeurIPS 2020, v. 33*. DOI: 10.48550/arxiv.2005.14165.

Cardoso, P. C., Maziero, E. G., Jorge, M. L. C., Seno, E. M., Di Felippo, A., Rino, L. H. M., Nunes, M. d. G. V., and Pardo, T. A. (2011). Cstnews-a discourse-annotated corpus for single and multi-document summarization of news texts in brazilian portuguese. In *Proceedings of the 3rd RST Brazilian Meeting*, pages 88–105. sn. Available at:`https://sites.icmc.usp.br/taspardo/rst2011-cardosoetal1.pdf`.

Collovini, S., Carbonel, T. I., Fuchs, J. T., Coelho, J. C., Rino, L., and Vieira, R. (2007). Summ-it: Um corpus anotado com informações discursivas visando a sumarização automática. In *Proceedings of the 5th Workshop in Information and Human Language Technology (NILC)*. Available at:`https://portulanclarin.net/repository/extradocs/Summit.pdf`.

de Vargas Feijó, D. and Moreira, V. P. (2018). Rulingbr: A summarization dataset for legal texts. In *International Conference on Computational Processing of the Portuguese Language*, pages 255–264. Springer. DOI: 10.1007/978-3-319-99722-3$_2$6.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., *et al*. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*. DOI: 10.48550/arXiv.2407.21783.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. In *Journal of Artificial Intelligence Research*, volume 22, pages 457–479. DOI: 10.1613/jair.1523.

Fabbri, A. R., Kryscinski, W., McCann, B., Xiong, C., Socher, R., and Radev, D. (2021). Summeval: Re-evaluating summarization evaluation. In *Transactions of the Association for Computational Linguistics (TACL)*, volume 9, pages 391–409. DOI: 10.1162/tacl$_{a0}$0373.

Fonseca, E. B., Antonitsch, A., Collovini, S., Amaral, D., Vieira, R., and Figueira, A. (2016). Summ-it++: an enriched version of the summ-it corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2047–2051. Available at:`https://aclanthology.org/L16-1324/`.

Galley, M. (2006). Skip-chain conditional random fields for ranking meeting utterances by importance. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 364–372. DOI: 10.7916/D8RV0X5W.

Garcia, G. L., Paiola, P. H., Garcia, E., Manesco, J. R. R., and Papa, J. P. (2024a). Gembode and phibode: Adapting small language models to brazilian portuguese. In *CIARP*. Accepted for publication. DOI: 10.1007/978-3-031-76607-7$_1$7.

Garcia, G. L., Paiola, P. H., and Papa, J. P. (2024b). Chatbode. Available at:`https://huggingface.co/recogna-nlp/internlm-chatbode-7b/`.

Gehring, J., Auli, M., Grangier, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1243–1252. DOI: 10.48550/arxiv.1705.03122.

Ghalandari, D. G., Hokamp, C., Pham, N. T., Glover, J., and Ifrim, G. (2020). A large-scale multi-document summarization dataset from the wikipedia current events portal. *arXiv preprint arXiv:2005.10070*. DOI: 10.48550/arXiv.2005.10070.

Gong, Y. and Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 19–25. DOI: 10.1145/383952.383955.

Gupta, S. and Gupta, S. (2019). Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65. DOI: 10.1016/j.eswa.2018.12.011.

Kupiec, J., Pedersen, J., and Chen, F. (1995). A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73, New York. ACM. DOI: 10.1145/215206.215333.

Kågebäck, M., Mogren, O., Tahmasebi, N., and Dubhashi, D. (2014). Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 31–39. DOI: 10.3115/v1/w14-1504.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*. DOI: 10.18653/v1/2020.acl-main.703.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Espanha. Association for Computational Linguistics. Available at:`https://aclanthology.org/W04-1013.pdf`.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. (2021). Gpt understands, too. *ArXiv*, abs/2103.10385. DOI: 10.1016/j.aiopen.2023.08.012.

Liu, Y., Jia, Q., and Zhu, K. (2022). Reference-free summarization evaluation via semantic correlation and compression ratio. In *Proceedings of NAACL-HLT*, pages 2109–2115, Seattle, WA, USA. Association for Computational Linguistics. DOI: 10.18653/v1/2022.naacl-main.153.

Liu, Y. and Lapata, M. (2019). Fine-tune bert for extractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 214–224. DOI: 10.48550/arxiv.1903.10318.

Marcu, D. (2000). *The theory and practice of discourse parsing and summarization*. MIT press. DOI: 10.7551/mitpress/6754.001.0001.

Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919. DOI: 10.18653/v1/2020.acl-main.173.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 404–411. Available at:`https://aclanthology.org/W04-3252.pdf`.

Nallapati, R., Zhai, F., Zhou, B., Gulcehre, C., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290. DOI: 10.18653/v1/k16-1028.

Nenkova, A. and McKeown, K. (2011). *Automatic summarization*. Now Publishers Inc, Boston. DOI: 10.1561/9781601984715.

Paiola, P. H., de Rosa, G. H., and Papa, J. P. (2022).

Deep learning-based abstractive summarization for brazilian portuguese texts. In Xavier-Junior, J. C. and Rios, R. A., editors, *BRACIS 2022: Intelligent Systems*, pages 479–493. Springer International Publishing, Cham. DOI: 10.1007/978-3-031-21689-3$_3$4.

Paiola, P. H., Garcia, G. L., Jodas, D. S., Correia, J. V. M., Sugi, L. A., and Papa, J. P. (2024). RecognaSumm: A novel Brazilian summarization dataset. In Gamallo, P., Claro, D., Teixeira, A., Real, L., Garcia, M., Oliveira, H. G., and Amaro, R., editors, *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*, pages 575–579, Santiago de Compostela, Galicia/Spain. Association for Computational Linguistics. Available at:`https://aclanthology.org/2024.propor-1.63.pdf`.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318. Association for Computational Linguistics. DOI: 10.3115/1073083.1073135.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. DOI: 10.48550/arxiv.1910.10683.

Sellam, T., Das, D., and Parikh, A. P. (2020). Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7881–7892. DOI: 10.18653/v1/2020.acl-main.704.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., Tafti, P., Hussenot, L., *et al*. (2024). Gemma. Available at:`https://www.kaggle.com/m/3301`.

Van Veen, D., Van Uden, C., Blankemeier, L., Delbrouck, J.-B., Aali, A., Bluethgen, C., Pareek, A., Polacin, M., Reis, E. P., Seehofnerova, A., *et al*. (2023). Clinical text summarization: adapting large language models can outperform human experts. *Research square*, pages rs–3. DOI: 10.21203/rs.3.rs-3483777/v1.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, □ and Polosukhin, I. (2017). Attention is all you need. In *NeurIPS*. DOI: h10.48550/arxiv.1706.03762.

Wang, P., Li, J., and Zhu, X. (2008). Multi-document summarization using sentence-based topic models. In *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 551–554. IEEE. DOI: 10.3115/1667583.1667675.

Widyassari, A. P. *et al*. (2022). Review of automatic text summarization techniques & methods. *Journal of King Saud University - Computer and Information Sciences*, 34(4):1029–1046. DOI: 10.1016/j.jksuci.2020.05.006.

Zhang, H., Yu, P. S., and Zhang, J. (2024). A systematic survey of text summarization: From statistical methods to large language models. DOI: 10.1145/3731445.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR. Online. DOI: 10.48550/arxiv.1912.08777.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations (ICLR)*. DOI: h10.48550/arxiv.1904.09675.

Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C. M., and Eger, S. (2019). Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 563–578. DOI: 10.18653/v1/d19-1053.

Zhao, W. *et al*. (2021). Calibrate before use: Improving few-shot performance of language models. *arXiv preprint arXiv:2102.09690*. DOI: 10.48550/arxiv.2102.09690.

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., and Huang, X. (2020). Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.552.

A One-Shot and Few-Shot Prompts

A.1 One-Shot Prompt

Você é um assistente responsável por resumir textos complexos de maneira clara e concisa. Sua tarefa é fornecer um resumo que capture os pontos principais e o contexto essencial. Abaixo está um exemplo de texto e seu respectivo resumo:

Exemplo 1: Texto: Tiago Tortellada CNN Em São Paulo Dados do Censo Demográfico 2022, divulgado nesta quarta-feira (28) pelo Instituto Brasileiro de Geografia e Estatística (IBGE), revelam que a população brasileira continua crescendo desde a última edição do levantamento, em 2010, tendo ultrapassado a barreira de 200 milhões de habitantes. Algumas capitais tiveram ganhos expressivos de residentes, como Boa Vista (RR), que teve aumento populacional de 45,4% em comparação a 2010, e Florianópolis (SC), que teve aumento de 27,5%. Já Manaus foi a cidade que teve o maior crescimento absoluto, de 14,5%, com 261 mil habitantes a mais, mais até que São Paulo e Brasília. Outras capitais, como Cuiabá (MT) e, João Pessoa (PB), também registraram aumento na população. Já Salvador (BA), Belo Horizonte (MG), Recife (PE) e Belém (PA), entre outras, apresentaram queda populacional. Leia Mais: O Censo 2022 é a 13ª operação do tipo realizada em território brasileiro, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE). Por lei, os censos são feitos com, no máximo, 10 anos de intervalo. Porém, devido à pandemia da Covid-19, a coleta de informações e formulação dos resultados foi adiada em 2020. Em 2021, segundo o instituto, também não foi realizado o levantamento devido ao “profundo corte orçamentário”, sendo finalmente aplicado em 2022. Os recenseadores do IBGE visitaram 106,8 milhões de endereços e 90,7 milhões de domicílios em 2022. Foram aplicados 62.388.143 questionários “básicos”, com 26 quesitos e tempo médio de 6 minutos; e 7.772.064 questionários "ampliados", com 77 quesitos e tempo médio de 16 minutos. Ao todo, 68.659.405 de entrevistas foram feitas presencialmente; 362.563 questionários foram preenchidos pela internet; e 412.725 entrevistas foram feitas por telefone. O instituto ressalta que os dados adquiridos por meio dos censos são utilizados, por exemplo, no planejamento social e econômico do país. Quase 70% das casas no Brasil não têm acesso a rede de esgoto, diz IBGE | CNN PRIME TIME data-youtube-width="500px" data-youtube-height="281px" data-youtube-ui="nacional" data-youtube-play="" data-youtube-mute="0" data-youtube-id="9GUcdHMZjAQ" Resumo: Censo 2022: Boa Vista e Florianópolis têm aumento populacional superior a 20% em 12 anos. Na capital de Roraima, número de habitantes aumentou 45,4% na comparação com 2010.

Texto: Este é o texto que queremos resumir. Resumo:

A.2 Few-Shot Prompt

Você é um assistente responsável por resumir textos complexos de maneira clara e concisa. Sua tarefa é fornecer um resumo que capture os pontos principais e o contexto essencial. Abaixo estão alguns exemplos de textos e seus respectivos resumos:

Exemplo 1: Texto: Tiago Tortellada CNN Em São Paulo Dados do Censo Demográfico 2022, divulgado nesta quarta-feira (28) pelo Instituto Brasileiro de Geografia e Estatística (IBGE), revelam que a população brasileira continua crescendo desde a última edição do levantamento, em 2010, tendo ultrapassado a barreira de 200 milhões de habitantes. Algumas capitais tiveram ganhos expressivos de residentes, como Boa Vista (RR), que teve aumento populacional de 45,4% em comparação a 2010, e Florianópolis (SC), que teve aumento de 27,5%. Já Manaus foi a cidade que teve o maior crescimento absoluto, de 14,5%, com 261 mil habitantes a mais, mais até que São Paulo e Brasília. Outras capitais, como Cuiabá (MT) e, João Pessoa (PB), também registraram aumento na população. Já Salvador (BA), Belo Horizonte (MG), Recife (PE) e Belém (PA), entre outras, apresentaram queda populacional. Leia Mais: O Censo 2022 é a 13ª operação do tipo realizada em território brasileiro, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE). Por lei, os censos são feitos com, no máximo, 10 anos de intervalo. Porém, devido à pandemia da Covid-19, a coleta de informações e formulação dos resultados foi adiada em 2020. Em 2021, segundo o instituto, também não foi realizado o levantamento devido ao "profundo corte orçamentário", sendo finalmente aplicado em 2022. Os recenseadores do IBGE visitaram 106,8 milhões de endereços e 90,7 milhões de domicílios em 2022. Foram aplicados 62.388.143 questionários "básicos", com 26 quesitos e tempo médio de 6 minutos; e 7.772.064 questionários "ampliados", com 77 quesitos e tempo médio de 16 minutos. Ao todo, 68.659.405 de entrevistas foram feitas presencialmente; 362.563 questionários foram preenchidos pela internet; e 412.725 entrevistas foram feitas por telefone. O instituto ressalta que os dados adquiridos por meio dos censos são utilizados, por exemplo, no planejamento social e econômico do país. Quase 70% das casas no Brasil não têm acesso a rede de esgoto, diz IBGE | CNN PRIME TIME data-youtube-width="500px" data-youtube-height="281px" data-youtube-ui="nacional" data-youtube-play="" data-youtube-mute="0" data-youtube-id="9GUcdHMZjAQ" Resumo: Censo 2022: Boa Vista e Florianópolis têm aumento populacional superior a 20% em 12 anos. Na capital de Roraima, número de habitantes aumentou 45,4% na comparação com 2010.

Exemplo 2: Texto: Damien Tarel, 28 anos, ficou conhecido após ter batido no rosto do presidente francês Emmanuel Macron, no dia 8 de junho. Condenado a uma curta pena em regime fechado, ele deixou a prisão no sábado (11) sem demonstrar arrependimento. "Não me arrependo deste ato", disse ele ao sair da penitenciária de Valence, no sudeste da França, onde cumpriu pena de três meses de reclusão. "Foi só um 'tapinha' e acho que Macron se recuperou muito bem", acrescentou Tarel para qualificar seu gesto. "Não se trata de uma surra como poderia ter havido nas manifestações dos coletes amarelos, onde o povo expressa o seu descontentamento", continuou o jovem. A agressão registrada em vídeo circulou pelas redes sociais. Nas imagens, Macron se aproxima de um grupo de pessoas que se encontravam atrás de barreiras de segurança para cumprimentá-las. Em seguida, o agressor agarra o antebraço do presidente e dá a bofetada. O incidente aconteceu nos arredores de uma escola de hotelaria na localidade de Tain-L'Hermitage, a 550 quilômetros de Paris. Na gravação também é possível ouvir uma pessoa pronunciar um antigo grito de guerra dos reis da França ("Montjoie Saint-Denis"), seguido da frase "Abaixo o governo Macron!". "Esta sentença pune um desrespeito intolerável à instituição", disse na época o promotor Alex Perrin. Além da prisão, a Justiça suspendeu os direitos civis de Damien Tarel por três anos, o que o impedirá de votar neste período. Ele também fica proibido de prestar concursos públicos para o resto da vida e não poderá deter armas de fogo nos próximos cinco anos. O juiz ainda ordenou que ele tenha acompanhamento psicológico. "Naquele dia, eu fui desafiar o presidente Emmanuel Macron e o que vi foram trabalhadores em coletes amarelos que estavam lá para expressar seu descontentamento, pessoas que trabalham muito, muitas vezes idosos, que eram retiradas pelas forças policiais pagas por seus impostos" e " isso me revoltou ", explicou o jovem desempregado. "O povo está amordaçado", acrescentou. Tarel ainda confirmou que compareceria às manifestações contra o passe sanitário imposto pelo governo francês para conter a epidemia de Covid-19, previstas para este sábado. "É minha prioridade depois de ser libertado da prisão, pelo menos pelo símbolo, aderir a este movimento que contesta a decisão do passe sanitário", concluiu. Veja os vídeos mais assistidos do G1 Resumo: Homem que deu tapa em Emmanuel Macron, da França, deixa prisão e afirma que não se arrependeu do crime. O homem cumpriu uma pena de três meses em regime fechado. Em junho, o agressor agarrou o antebraço do presidente da França e deu uma bofetada.

Exemplo 3: Texto: O deputado federal Márcio Macedo (PT-SE) assumiu nesta segunda-feira (2) o cargo de ministro-chefe da Secretaria-Geral da Presidência da República. Durante a cerimônia de transmissão de cargo, Macedo disse que deseja, por meio da Secretaria-Geral, aproximar a população da presidência. "Faremos da Secretaria-Geral um elo entre a população e a Presidência da República. Estejam certos que vou acordar todos os dias para alimentar esse trabalho." A Secretaria-Geral da Presidência é um órgão que integra a Presidência da República e tem o objetivo de ajudar o presidente na condução estratégica de governo. Com status de ministério, a pasta funciona no Palácio do Planalto, onde também estão abrigados a Casa Civil, a Secretaria de Relações Institucionais (SRI) e o Gabinete de Segurança Institucional (GSI), além dos gabinetes do presidente e do vice-presidente da República. No novo desenho ministerial do governo de Luiz Inácio Lula da Silva (PT) , a Secretaria-Geral da Presidência da República ficou responsável por: coordenar e articular as relações políticas do governo federal com os diferentes segmentos da sociedade civil e da juventude;coordenar a política e o sistema nacional de participação social;criar, implementar, articular e monitorar instrumentos de consulta e participação popular nos órgãos governamentais de interesse do Executivo;cooperar com os movimentos sociais na articulação das agendas e ações que fomentem o diálogo, a participação social e a educação popular;fortalecer e articular os mecanismos e as instâncias democráticas de diálogo e a atuação conjunta entre a administração pública federal e a sociedade civil;debater com a sociedade civil e com o Executivo iniciativas de plebiscitos e referendosPerfil Márcio Macedo nasceu no município de Esplanada, na Bahia. Ingressou na Universidade Federal de Sergipe em 1989, onde concluiu sua graduação em Ciências Biológicas e, depois, o mestrado em Desenvolvimento e Meio Ambiente. Filiado ao PT, ocupou o cargo de presidente dos diretórios municipal de Aracaju e estadual de Sergipe do partido. Entre 2007 e 2010, foi secretário do Meio Ambiente e dos Recursos Hídricos de Sergipe na gestão do então governador Marcelo Déda. Em 2015, ocupou o cargo de tesoureiro do PT, permanecendo na função até 2020. Atualmente, é um dos vice-presidentes nacionais do Partido dos Trabalhadores. Foi ainda secretário municipal de Participação popular de Aracaju e superintendente do Ibama em Sergipe. Márcio lançou sua candidatura a deputado federal em 2010, sendo eleito nas eleições daquele ano com 58.782 votos. Durante o mandato, na legislatura de 2011 a 2014, chegou a ser vice-líder do PT, além de ter presidido a Comissão de Mudanças Climáticas do Congresso. Em 2018, terminou as eleições com uma suplência para o cargo de deputado federal. Em abril de 2022, tomou posse na Câmara dos Deputados. Na campanha de 2022, foi tesoureiro da chapa de Luiz Inácio Lula da Silva. Resumo: Márcio Macedo assume Secretaria-Geral da Presidência da República. Na campanha de 2022, deputado federal foi tesoureiro da chapa de Lula. Com status de ministério, pasta tem o objetivo de ajudar o presidente na condução estratégica do governo.

Exemplo 4: Texto: A mensagem, de 12 de abril deste ano, foi enviada pelo presidente ao então ministro após a publicação de uma reportagem que registrou a opinião dada por Moro em uma videoconferência. O texto consta dos documentos anexados ao inquérito da Polícia Federal que apura a suposta interferência do presidente da República na Polícia Federal. No dia 12 de abril, um domingo, o jornal ”Valor Econômico” noticiou a participação de Moro em uma videoconferência de uma empresa de investimentos. Nessa videoconferência, o ministro explicou que a portaria que trata da emergência sanitária relativa ao novo coronavírus permitiria que a polícia atuasse até coercitivamente para o cumprimento das regras. Cerca de uma hora após a publicação da reportagem, Bolsonaro enviou um link do texto para o ministro. E disse: ”Se esta matéria for verdadeira: Todos os ministros, caso queira (sic) contrariar o PR, pode fazê-lo, mas tenha dignidade para se demitir. —Aberto para a imprensa.” Cinco minutos depois, Moro respondeu. ”O que existe eh(sic) o art 268 do CP. Não falei com imprensa.” O artigo 268 do Código Penal define o crime de infração de medida sanitária, que consiste em ”infringir determinação do poder público, destinada a impedir introdução ou propagação de doença contagiosa”. O crime é punível com até um ano de detenção e multa, podendo ser agravada se o infrator for profissional de saúde. A citação à mensagem é feita pela Polícia Federal num relatório de análise de material apreendido na investigação — as mensagens do celular do ex-ministro Sergio Moro foram analisadas pelos investigadores. O relatório diz que seriam destacados ”tão somente os trechos considerados relevantes para o contexto investigativo”. Não há, entretanto, nenhuma avaliação quanto à relação entre esta mensagem e a investigação de possível interferência do presidente da República na Polícia Federal. VÍDEOS: as notícias mais vistas do G1 Resumo: Inquérito da PF relata mensagem de Bolsonaro a Moro: ’Tenha dignidade para se demitir’ Mensagem foi enviada em abril, após então ministro da Justiça dizer que prisão por infração sanitária era possível. Registro consta de inquérito sobre suposta interferência na PF.

Texto: Este é o texto que queremos resumir. Resumo:

B Examples of Generated Summaries

- **Source text**: O presidente dos Estados Unidos, Joe Biden, sancionou nesta quinta-feira (16) o projeto de lei que eleva o limite da dívida norte-americana, informou a Casa Branca em comunicado, evitando o que teria sido um calote sem precedentes. O Congresso dos EUA aprovou na terça-feira (16) o aumento da autoridade de empréstimos do governo federal norte-americano em US$ 2,5 trilhões, para cerca de US$ 31,4 trilhões, enviando a proposta para aprovação de Biden.
- **Reference summary**: Biden torna lei projeto que aumenta teto da dívida dos EUA. Congresso aprovou na terça-feira o aumento da autoridade de empréstimos do governo federal norte-americano em US$ 2,5 trilhões.
- **ptt5_xlsum**: O governo dos Estados Unidos anunciou que o Congresso dos Estados unidos aprovou um projeto de lei que aumenta a autoridade da dívida americana. A medida foi anunciada nesta quinta-feira. A decisão foi anunciada pelo governo americano. A informação foi divulgada pela Casa Branca.
- **ptt5_temario**: O presidente dos Estados Unidos, Joe Biden, sancionou o projeto de lei que eleva o limite da dívida norte-americana. O Congresso dos EUA aprovou o aumento da autoridade de empréstimo do governo norte-americano em US$ 2,5 trilhões, para cerca de US$ 31,4 trilhões.
- **chatbode**: O presidente Biden aprovou o projeto de lei que aumenta o limite da dívida norte-americana, evitando um calote sem precedentes. O Congresso aprovou o aumento da autoridade de empréstimos do governo federal norte-americano em US$ 2,5 trilhões, para cerca de US$ 31,4 trilhões, enviando a proposta para aprovação de Biden.

- **gembode**: O presidente dos EUA, Joe Biden, sancionou o projeto de lei que aumenta o limite da dívida do governo federal norte-americano em US$ 2,5 trilhões, evitando um calote sem precedentes.
- **llama**: O presidente Joe Biden sancionou o projeto de lei que aumenta o limite da dívida dos EUA, evitando um calote sem precedentes. O Congresso aprovou o aumento da autoridade de empréstimos em US$ 2,5 trilhões, elevando o limite para US$ 31,4 trilhões.
- **gemma**: O presidente dos Estados Unidos, Joe Biden, sancionou o projeto de lei que eleva o limite da dívida norte-americana, evitando um calote sem precedentes. O Congresso dos EUA aprovou o aumento da autoridade de empréstimos do governo federal norte-americano em US$ 2,5 trilhões, enviando a proposta para aprovação de Biden.