TeenyTinyLlama: a pair of open-source *tiny* language models trained in Brazilian Portuguese

Nicholas Kluge Corrêa[1,2]   **Sophia Falk**[1]   **Shiza Fatimah**[1]   **Aniket Sen**[1]

Nythamar de Oliveira[2]

[1]University of Bonn
[2]Pontifical Catholic University of Rio Grande do Sul

Abstract

Large language models (LLMs) have significantly advanced natural language processing, but their progress has yet to be equal across languages. While most LLMs are trained in high-resource languages like English, multilingual models generally underperform monolingual ones. Additionally, aspects of their multilingual foundation sometimes restrict the byproducts they produce, like computational demands and licensing regimes. In this study, we document the development of open-foundation models tailored for use in low-resource settings, their limitations, and their benefits. These are the *TeenyTinyLlama* pair: two compact models for Brazilian Portuguese text generation. We release them under the permissive Apache 2.0 license on GitHub and Hugging Face for community use and further development.

1 Introduction

Large language models have radically changed the field of natural language processing (NLP) with their exceptional ability to perform downstream tasks after being trained on vast amounts of data in a self-supervised learning regime. Under this paradigm, transformer-based models like BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), mT5 (Xue et al., 2020), and the whole family of

Preprint, work in progress.

GPT-style models (Radford et al., 2019; Black et al., 2022; Workshop et al., 2022; Biderman et al., 2023; Gunasekar et al., 2023; Almazrouei et al., 2023; Touvron et al., 2023b; Luo et al., 2023; Jiang et al., 2024), have become the foundation for many NLP applications and research areas. While part of the field is still pushing the search for new architectures, with innovations like the RWKV (Peng et al., 2023) and state space models (Gu and Dao, 2023) promising new directions for research, the majority of LLM research is still focused on the scaling of model size, training data, and the general efficiency and capabilities of transformer-based LLMs (Vaswani et al., 2017).

Despite the tremendous success of the field, progress has yet to be made equally regarding all languages. Hence, another trend in current NLP research involves the expansion of language domains with which such systems can interact. Current practices to tackle our linguistic multitude involve either training singular models in multiple languages (Conneau et al., 2019; Lin et al., 2021b; Workshop et al., 2022; Shliazhko et al., 2022) or fine-tuning foundational models trained on multi-linguistic corpora to become monolingual or more proficient when working with low-resource languages (Eisenschlos et al., 2019; Guillou, 2020; Alabi et al., 2022; Pires et al., 2023; Nguyen et al., 2023b; Lankford et al., 2023; Zhao et al., 2024).

However, most multilingual models available today still have a disproportional performance across languages due to the imbalance of training data, where usually high-resource languages, like English, represent the majority of such corpora, which creates user dissatisfaction with multilingual model's capabilities on non-English languages. Meanwhile, fine-tuned byproducts of multilingual models sometimes end up being restricted by the conditions imposed by the foundation used, like high computational costs for training and inference, which restrain adoption in low-resource settings, besides licensing regimes that prevent free use and open-source development. Factors like these highlight the necessity of building the foundations for monolingual LLMs for low-resource languages (Martin et al., 2020; Souza et al., 2020; Scheible et al., 2020; Antoun et al., 2021; Nagoudi et al., 2021; Gutiérrez-Fandiño et al., 2021; Ko et al., 2023; Rodrigues et al., 2023).

This study follows the trend of developing LLMs tailored for low-resource regimes (Gunasekar et al., 2023; Zhang et al., 2024; Team, 2024). In it, we sought to explore the challenges of developing LLMs in such settings. For this, we chose Brazilian Portuguese as our target language. To our knowledge, only a few LLMs for text generation were trained or fine-tuned to be proficient in Brazilian Portuguese and are available to the general public. Meanwhile, even fewer are available with permissive licenses and open-source code development. The models we developed, the *TeenyTinyLlama* pair (TTL), were created to help democratize LLMs for low-resource languages and the open-source community in general, providing a simple and extensible implementation for LLM pre-training and fine-tuning at a small (< 2B parameters) scale.

2 Related Works

As already stated, multilingual models like BLOOM (Workshop et al., 2022), mGPT (Shliazhko et al., 2022), and XGLM (Lin et al., 2021b) usually do not have satisfactory performance on low-resource languages, especially when compared to monolingual models. To overcome this, much of the community repurposes models trained on multilingual corpora to create mono-linguistic models for a target language via supervised fine-tuning (SFT), or, as some call it (Pires et al., 2023; Larcher et al., 2023), extended pre-training.

Sometimes, this strategy even works on models not trained on multilingual datasets. For example, one of the first text-generation models for Brazilian Portuguese to appear to the general public was Pierre Guillou GPorTuguese-2 in 2020 (Guillou, 2020), which coincided with the release of one of the first LLM natively trained in Brazilian Portuguese (BERTimbau) (Souza et al., 2020). GPorTuguese-2 is a byproduct of fine-tuning GPT-2 small (Radford et al., 2019) on the Portuguese portion of the Wikipedia dataset (Wikimedia Foundation, 2024) while also modifying the structure of the original model, giving it a new byte-pair encoding (BPE) tokenizer and repurposing the joint embeddings from the original model. While capable of generating fluent text in Brazilian Portuguese, GPorTuguese-2 fails to achieve a perplexity score on par with GPT-2 small, as documented by OpenAI, which is not surprising, given the limited hours of training ($\approx$ 30 hours of training) it received. Regardless, this pioneering work is available to all users under a permissive MIT License (as all GPT-2 models).

However, since the proposition of the GPT-2 architecture, many advances in transformer design have been made, and currently, more modern models are the default choice for engineers (Shoeybi

et al., 2019; Black et al., 2022; Jiang et al., 2023). Perhaps one of the most significant contributions to the open source community in 2023 was the release of the Llama 2 architecture (Touvron et al., 2023b). Being the successor of Llama (Touvron et al., 2023a), it brings many improvements that make training and inference of transformer-based language models more efficient, like the use of grouped query attention (Ainslie et al., 2023), better sub-layer normalization techniques (Zhang and Sennrich, 2019), changing ReLU activation's by SwiGLU (Shazeer, 2020), and the use of rotary positional embeddings instead of positional ones (Su et al., 2021), besides being trained on a massive pre-training corpus, even beyond what is estimated to be optimal by the scaling Chinchilla laws (Hoffmann et al., 2022).

Currently, there is an entire ecosystem of Llama-based models being released on a Cambrian explosion rate (Geng and Liu, 2023; Luo et al., 2023; Bi et al., 2024; Zhang et al., 2024; NousResearch, 2024; Roziere et al., 2023), with other open-architectures like Mistral (Jiang et al., 2023) also being heavily used by the open source community. Yet, it is from the LLama models that most of the other Brazilian Portuguese models come from. Three byproducts of fine-tuning (or extending the training of) Llama 1/2 with Brazilian Portuguese corpora are Bode (Garcia et al., 2024), Sabiá (Pires et al., 2023), and Canarim (Maicon Domingues, 2023a).

Bode is a low-rank adaptation (LoRA) of Llama 2 fine-tuned with a translated version of the Alpaca dataset (Taori et al., 2023), which contains 52K instruction-following demonstrations generated by text-davinci-003 (Ouyang et al., 2022). Bode is offered as LoRA adapters for Llama 7B and 13B while having a fine-tuned version of Llama 7B without using LoRA or other parameter-efficient fine-tuning techniques. In short, Bode only went through a fine-tuning phase, and the results of the model capacity are documented in their paper (Garcia et al., 2024). According to these, Bode stays within the performance of models that underwent a similar training regime, like Cabrita (Larcher et al., 2023). It is also on par with its base model and even surpasses it on specific evaluations performed by the authors. However, like all derivatives of Llama models, Bode is licensed under the Llama 2 Community License Agreement, which is not as permissive as Apache 2.0, MIT, or commercial versions of the Creative Commons licenses.

Meanwhile, Sabiá models are fine-tuned versions of GPT-J (Wang and Komatsuzaki, 2021), and Llama trained on a filtered portion of the ClueWeb 2022 dataset (Overwijk et al., 2022), which equates to 7.3–7.8 billion tokens, according to GPT-J and Llama tokenizers, respectively. The outcomes of this extended training process are Sabiá-7B, 65B (both derivatives of Llama), and Sabiá-J (using GPT-J as a base). According to the authors, their evaluations show that Sabiá-65B outperforms Llama 2-65B and GPT-3.5-turbo on the Portuguese Evaluation Tasks (Poeta) benchmark (a set of tests gathered by the authors of the Sabiá paper). However, Sabiá-65B and Sabiá-J are not available to the public,[1] while Sabiá-7B, just like the Bode models, was released with a Llama 2 license. Also, none of these models' training, evaluation, or fine-tuning source codes have been released to the community.

Regarding Canarim, there is not much to be said except that it is also a Llama 2 licensed model that underwent an extended training process on 16 billion tokens from a Portuguese subset of Common Crawl (2023) and was further fine-tuned on two datasets, one for instruction-tuning (Maicon Domingues, 2023b) and the other for open-ended question answering with a focus on the ENEM exams. Again, the models are also under the licensing regime of the Llama model's community license. For the interested reader, there are many other examples of these types of models on repositories like Hugging Face (Leonardo Souza, 2024; Bruno Henrique, 2024a,b; Nicolas de Camaret, 2024).[2]

Finally, there are the Cabrita models (Larcher et al., 2023). Unlike the work done by Garcia et al. (2024), Pires et al. (2023), and Maicon Domingues (2023a), Cabrita is a byproduct of fine-tuning OpenLLaMA 3B (Geng and Liu, 2023), which comes with an Apache 2.0 license. Also, Cabrita models have a modified tokenizer, unlike Sabiá, Bode, and Canarim, which directly repurposed the original Llama tokenizers. Cabrita extended training was performed on a subset of the mC4 dataset (Raffel et al., 2019), using $\approx$ 7 billion tokens. Cabrita 3B, like its base model, is available under an Apache 2.0 license.

---

[1]Even though GPT-J is available under an Apache 2.0 License.

[2]That is, LLMs repurposed for other languages via full or LoRA/PEFT fine-tuning not accompanied by a paper or report.

To our knowledge, open-source LLMs for text generation pre-trained solely in Brazilian Portuguese are inexistent, nor have the above projects open-sourced their training and evaluation methods for reproducibility and further community development. At the same time, all models cited, based on billion-sized transformers, require non-trivial computational resources to use, adapt, and reproduce in low-resource settings. Hence, this study aimed to produce a pair of compact LLMs in Brazilian Portuguese by pre-training them from scratch (the TTL pair), tailored for (and produced by) a low-resource environment. The rest of this work documents our development, experiments, and results.

3 Pre-Training

In this section, we will describe how we designed the training of our TTL models. This study was performed with a closed budget of 500,00 USD, forcing us to make many developmental decisions to reduce costs and optimize our pre-training runs. This low-resource setting also influenced the size of our models that, as already shown by Pires et al. (2023), can range from 9,000 to 80,000 USD when training billion parameter-sized models on a dataset similar to what we have gathered. Meanwhile, training models with trillion tokens, even in small settings, is beyond what we could finance and what is possible to accumulate with available Brazilian Portuguese text datasets. However, scaling down allowed us to choose a range of sizes where, according to scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022), our limited budget was enough to pay for the computing necessary to pre-train our models and evaluate them.

3.1 Sizing Up Models and Datasets

While empirical evidence seems to point to the fact that existing scaling laws (Hoffmann et al., 2022) may not provide accurate predictions in situations where smaller models are trained for more extended periods (Touvron et al., 2023b; Zhang et al., 2024), in this study, we choose to use the Hoffmann et al. (2022) scaling laws, like done by Dey et al. (2023), to estimate the size of models. Even though extrapolating such boundaries might benefit smaller models, we did not have the budget (or tokens) to sustain longer runs.

According to Hoffmann et al. (2022), we can model language modeling loss, L, as a function of model size N (the number of parameters) and training dataset size D (the number of tokens):

$$L(N, D) = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + E.$$

Where $A = 406.4$, $B = 410.7$, $E = 1.69$, $\alpha = 0.32$, and $\beta = 0.28$ are parameters estimated by the authors after fitting a regression model to a dataset of 400 language model training runs. In their paper, Hoffmann et al. (2022) present estimations for dataset size for many model sizes. With 70B parameters, Chinchilla requires 1.4T tokens according to these laws, which equates to roughly 20 tokens per parameter. Based on this average, we estimated an optimal dataset size for two models: 3.5 and 9.5 million tokens for 160 and 460 million parameter models, respectively. We considered these to be fair sizes for this project, given that we would be able to train them without requiring much computing while, at the same time, the token count was still within a manageable range, i.e., something we could gather by relying on open-source datasets.[3]

---

[3]We also experimented with extended training and embedding transplant, like in other works (Garcia et al., 2024; Pires et al., 2023; Maicon Domingues, 2023a). In one of our initial explorations, we recycled the GPorTuguese-2 tokenizer and embedding layer, using them as a replacement for the original tokenizer and embedding weights (which are of the same dimension, i.e., 768) used by Pyhtia-160m (Biderman et al., 2023). We then performed a test training run of 100,000 steps using the same hyperparameters and settings later described in our work. In our experiments, the loss curves exhibit significant variance, with sudden increases in loss, suggesting a lack of smooth convergence during the first quarter of training. This has led us further into favoring the idea of pre-training a model from scratch, which, in the end, allowed us to train our TTL pair smoothly from beginning to end.

3.2 Pre-Training Dataset

We consider English to be a high-resource language because, with datasets like the Pile (Gao et al., 2020), RedPajama (Computer, 2023), the ROOTS corpus (Laurençon et al., 2022), the Stack (Kocetkov et al., 2022), UltraChat (Ding et al., 2023), etc., one can easily have access to trillions of high-quality, domain-specific tokens. Currently, most available tokens in Brazilian Portuguese come from datasets like BrWaC (Wagner Filho et al., 2018), ClueWeb22 (Overwijk et al., 2022), Wikipedia (Wikimedia Foundation, 2024), OSCAR (Abadji et al., 2022), and other byproducts of massive web crawling that require considerable filtering and pre-processing (Ortiz Su'arez et al., 2019; Ortiz Su'arez et al., 2020; Wenzek et al., 2020; Conneau et al., 2020).

Studies like the ones performed by Xue et al. (2023) and Muennighoff et al. (2023) explore the challenges of training LLMs under token crises, i.e., settings where the amount of data available is constrained. Both studies mainly focus on the downsides of repeating data during training runs, given that training language models with fresh data seem to have beneficial outcomes (Lee et al., 2021). According to Muennighoff et al. (2023), under token crisis scenarios, training with up to 4 epochs of repeated data yields minor changes to loss compared to unique data. After this mark, increased repetition yields less performance, eventually decaying to zero. With this in mind, we aimed to build a dataset allowing training runs to be extended up and pass the optimal range without reaching the 4 epoch mark.

Hence, the first portion of our dataset comprises a concatenation of open-source Brazilian Portuguese datasets. These include: Wikipedia (Wikimedia Foundation, 2024), CulturaX (Nguyen et al., 2023a), OSCAR (Ortiz Su'arez et al., 2019; Ortiz Su'arez et al., 2020; Abadji et al., 2022), Common Crawl (Wenzek et al., 2020; Conneau et al., 2020), and ROOTS (Laurençon et al., 2022) datasets. As a filtering step, we also utilized some of the filters used in Rae et al. (2021), besides using a fine-tuned BERTimbau (Souza et al., 2020) to exclude samples classified above a pre-defined toxicity threshold.[4] This first portion equates to 4.1 billion tokens that occupy approximately 50 GB of memory. We call this first portion Pt-Corpus.

The second portion of our dataset was inspired by the many studies that show that models fine-tuned with demonstrations of instruction-following behavior perform better in many downstream tasks Askell et al. (2021); Ouyang et al. (2022); Bai et al. (2022); Chung et al. (2022); Touvron et al. (2023b); Shen et al. (2023), leading us to experiment with the inclusion of such type of data as part of a pre-training corpus. For this, we utilized the following datasets: Instruct-PTBR (Carlo Moro, 2024), Gpt4all-J (Pablo Filetti Moreira, 2024), Bactrian-X (Li et al., 2023), Dolly 15K (Conover et al., 2023), and CosmosQA (Huang et al., 2019), many of which are translated versions of English native datasets. The concatenation of both portions is what we call Pt-Corpus-Instruct. 60% of this corpus is plain Brazilian Portuguese text (e.g., books, articles, blogs, etc.), while 40% demonstrate instruction-following behavior. Pt-Corpus-Instruct equates to approximately 6.2 billion tokens that generate a memory footprint of 80 GB. Using Muennighoff et al. (2023) work, we estimated that this amount would be sufficient to train our models to pass the optimal point and up to 1 billion parameters if desired.

3.3 Tokenization

As pointed out by Cui et al. (2023) and Larcher et al. (2023), one of the obstacles related to adapting LLMs to new low-resource languages is the recycling of the tokenizer. For example, since Llama 2 was trained on a primarily English corpus, its tokenizer requires more tokens to encode non-English languages, shattering the information that could be already encoded into words or sub-units of words, as it does with much of the English vocabulary. However, one cannot merely exchange the tokenizer of a language model without some surgical adaptation of the embedding layer, as done by Guillou (2020), given that this exchange would break the learned mapping between tokens and embeddings.

Since the model architecture we adopt in this study is Llama 2, we trained a Sentencepience tokenizer (Kudo and Richardson, 2018) to make our model compatible with the ever-growing Llama ecosystem. We trained our tokenizer on 2 million text samples from our dataset, with a vocabulary size of 32K tokens. To test its efficiency, we performed the same test used by Larcher et al. (2023) to access their tokenizer, where we counted the number of tokens required to encode 7400 words. According to the

---

[4]This model is available on Hugging Face: https://huggingface.co/nicholasKluge/ToxicityModelPT.

results in Table 1, our tokenizer shows a 66% improvement in efficiency compared to the original Llama 2 tokenizer, allowing for a more efficient way to encode Brazilian Portuguese text.

Table 1: Tokenizer Efficiency

Model Tokenizer	nº of tokens	Vocabulary Size
TTL	**9,937**	32,000
GPorTuguese-2	9,959	50,257
BERTimbau	11,006	29,794
Cabrita-3B	11,488	52,000
Sabiá-7B	14,813	32,000

We utilized our tokenizer to encode our dataset into sequences of 2048 tokens. The raw text datasets and their tokenized versions are available for download on Hugging Face.

3.4 Architecture

As done by Zhang et al. (2024), we used a decoder-only Transformer model (Vaswani et al., 2017) based on Llama 2 (Touvron et al., 2023b) as the basis for our models. The dimensions of our models are documented in Table 2.

Table 2: TTL model's architecture

Size	Hidden size	Intermediate size	Context length	Heads	Layers	Vocab size
160M	768	3,072	2,048	12	12	32,000
460M	1,024	4,096	2,048	16	24	32,000

Our models have all the implementations that the Llama 2 architecture benefits from, i.e., grouped query attention (Ainslie et al., 2023), root mean square layer normalization (Zhang and Sennrich, 2019), SwiGLU activation's (Shazeer, 2020), and RoPE embeddings (Su et al., 2021). While our 160 million parameter model uses 12 attention heads paired with 12 key-value heads, the 460 million parameter version uses 16 attention and key-value heads.

3.5 Training

We created all of our code implementations using the libraries tied to the Hugging Face ecosystem, i.e., Transformers (Wolf et al., 2020), Datasets (Lhoest et al., 2021), Tokenizers (HuggingFace, 2019), and Accelerate (Gugger et al., 2022), which allow for easy reproducibility, adaptation, and further scaling. Our training and evaluation scripts follow a standard PyTorch structure (Paszke et al., 2019), while we utilized CodeCarbon (CodeCarbon, 2019) and Weights & Biases (Weights&Biases, 2017) for tracking our experiments.

Regarding hardware, we were limited to using a single NVIDIA A100-SXM4-40GB. To optimize its use, we performed several experiments to find the least costly training configuration regarding computing time and resource consumption, mainly regarding GPU memory utilization and tokens per second throughput. In these experiments, we explored the use of different mixed precision strategies and math modes (fp32, fp16, bf16, tf32), gradient accumulation steps, gradient checkpoints (Chen et al., 2016), the use of FlashAttention (Dao et al., 2022; Dao, 2023), different types of optimizers (Kingma and Ba, 2014; Loshchilov and Hutter, 2017; Shazeer and Stern, 2018; Dettmers et al., 2022), and the use of data preloading versus data streaming. Ultimately, we arrived at the following training configurations, showcased in Table 3, which produced a throughput of up to 29,491 tokens per second during training and $3\times$ that during inference on an Ampere GPU. The hyperparameters related to the optimizer and learning rate scheduler were based on the documentation of other open-source LLMs of similar size (Zhang et al., 2022; Workshop et al., 2022; Biderman et al., 2023).

This setting increased token throughput during training three times and in evaluation six times compared to using float32 precision, no tf32 mode, and a vanilla attention mechanism. The training of

Table 3: TTL model's training configuration

TTL-160m

Key	Value
tokens per batch	8,192
total training steps	458,000
gradient accumulation steps	1
optimizer	AdamW
learning rate	$6.0 \times e^{-4}$
adam epsilon	$1.0 \times e^{-8}$
adam beta 1	0.9
adam beta 2	0.999
weight decay	0.01
scheduler type	cosine
warmup steps	5,000
gradient checkpointing	False
mixed precision	bfloat16
tf32	True
flash attention 2	True

TTL-460m

Key	Value
tokens per batch	8,192
total training steps	1,200,000
gradient accumulation steps	2
optimizer	AdamW
learning rate	$3.0 \times e^{-4}$
adam epsilon	$1.0 \times e^{-8}$
adam beta 1	0.9
adam beta 2	0.999
weight decay	0.01
scheduler type	cosine
warmup steps	10,000
gradient checkpointing	False
mixed precision	bfloat16
tf32	True
flash attention 2	True

The full details are available in our GitHub repository.

TTL-160m took approximately 36 hours (1.5 days), while the training of our 460 million parameter version took 280 hours (11.5 days).[5]

During training, we saved several checkpoints for each model between an interval of 22,000 steps for TTL-160 and 25,000 for TTL-460m, resulting in 20 and 48 intermediate checkpoints, respectively. All checkpoints were saved along with the current state of their optimizer and scheduler, allowing our models to resume training at any checkpoint desired or for others to use these checkpoints as a starting point for further training or fine-tuning. At the same time, we measured, for each checkpoint, their estimated energy consumption and carbon emissions, which we used to compare with our model evaluation scores. We evaluated our models every 100,000 steps with a sample size corresponding to approximately 1% of the training dataset. All models trained and checkpoints are available on Hugging Face.

4 Results

4.1 Evaluations

During our training runs, both models showed consistent convergence. At no point did our evaluation curves show signs of overfitting or saturation.[6] In the case of our 460m parameter model, we intentionally trained past the optimal point by approximately 75,000 steps to assess if there were any signs of saturation, but our evaluations consistently gave better results. We hypothesize that our models are under-trained but can improve if further trained to pass the Chinchilla optimal range. As suggested by Touvron et al. (2023b) and Zhang et al. (2024), perhaps the scaling laws proposed Hoffmann et al. (2022) are indeed ill-suited to estimate the performance of small language models. In Figure 1, we present the learning curves of our TTL pair.

Like many other low-resource languages, Brazilian Portuguese does not possess a standardized set of benchmarks that can be run to create model comparisons. Despite the introduction of the Poeta benchmark (Pires et al., 2023), as of the writing of this work, it still needs to come with a reproducible implementation of its use, which is a problem also reported by Larcher et al. (2023). On the other

---

[5]Even though not used in this study, given the limitation of the number of GPUs we were able to use, our code implementation supports all parallel features from Accelerate, like distributed training on multiple GPUs, multiple nodes, and plugins to other distributed training libraries, like DeepSpeed (Rajbhandari et al., 2020), FSDP (Ott et al., 2021), and Megatron-LM (Shoeybi et al., 2019).

[6]By saturation, we refer to the phenomenon where a model stops improving after a certain threshold of ingested tokens, probably due to the model size itself (Biderman et al., 2023).

Figure 1: Learning Curves for the TTL pair

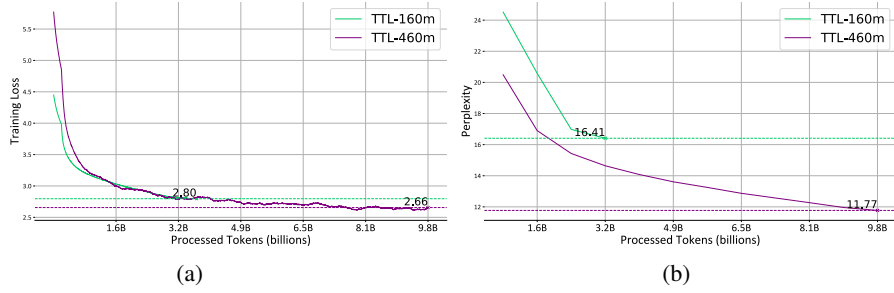(a)   (b)

Figure 1: Plot (a) shows the training loss of our TTL pair over their runs. TTL-460 was trained 75,000 steps passed our estimations (based on Hoffmann et al. (2022). Both models show consistent convergence, and we speculate that they could be trained for longer with a significant increase in improvements, as demonstrated by Biderman et al. (2023), on the training of even smaller models using over 300B tokens. Plot (b) shows the perplexity scores of our TTL pair measured at every 100,000 steps (8.1 million tokens). We were surprised that, even when trained on a comparatively smaller dataset, our models achieved a perplexity score similar to the results shown by Radford et al. (2019) for models of similar size. We attribute this to the superiority of the Llama 2 architecture compared to GPT-2, which, given the constant advances in the field, comes with several improvements that make training large neural networks more efficient.

hand, benchmarks like the Language Model Evaluation Harness (Gao et al., 2021) permit a common way to test generative language models on few-shot evaluations. However, even though the evaluation harness possesses a plethora of benchmarks for testing, in its current implementation, these are not available for languages like Brazilian Portuguese.

To bypass these problems and perform an evaluation that can be reproduced and further expanded, we relied on the work of Lai et al. (2023), which translated four benchmarks from the original evaluation harness to 29 languages in a commendable effort. These are:

- ARC-Challenge: a multiple-choice question-answering dataset containing questions from early grades science exams (Clark et al., 2018).

- HellaSwag: a multiple choice dataset for evaluating grounded commonsense inference (Zellers et al., 2019).

- MMLU: a benchmark that covers 57 subjects across STEM, humanities, social sciences, and more, measuring the performance of models on various natural language tasks (Hendrycks et al., 2020).

- TruthfulQA: a benchmark comprised of several questions, spanning 38 topics, that assess the model's tendency to replicate commonly believed falsehoods (Lin et al., 2021a).

For comparison purposes, we evaluated models categorized as within the same size range as the TTL pair on these benchmarks. Our results are in Table 4. Also, given that we saved several checkpoints of our models, this allows us to explore the evolution of LLMs capabilities as they are trained, as done by Biderman et al. (2023) and Zhang et al. (2024). In Figure 2, we display the evolution of TTL-460m on the ARC-Challenge over 13 checkpoints.

To further evaluate the downstream capabilities of our models, we decided to employ a basic fine-tuning procedure of our TTL pair on a subset of tasks of the Poeta benchmark (Pires et al., 2023). Those are tasks involving toxicity detection (Vargas et al., 2022), textual entailment (Rodrigues, 2023; Real et al., 2020), sentiment analysis (Maas et al., 2011), and a text classification (Zhang et al., 2015). We apply the same procedure for comparison purposes on both BERTimbau models (Souza et al., 2020), given that they are also LLM trained from scratch in Brazilian Portuguese and have a similar size range to our models. Given their bidirectional nature, encoder-only transformers are usually superior in tasks like text classification, SQUAD-style Q&A, and named entity recognition. However, we argue that we can still use these comparisons, especially if they are made in a standardized fashion, to assess if our pre-training runs produced LLM capable of producing good results ("good" here means "close to BERTimbau") when utilized for downstream applications. In this round of

Table 4: Performance on the Language Model Evaluation Harness (Gao et al., 2021).

	ARC	HellaSwag	MMLU	TruthfulQA	Average
Pythia-410m	24.83*	**41.29***	25.99*	40.95*	33.26
TTL-460m	**29.40**	33.00	**28.55**	41.10	33.01
Bloom-560m	24.74*	37.15*	24.22*	42.44*	32.13
Xglm-564M	25.56	34.64*	25.18*	**42.53**	31.97
OPT-350m	23.55*	36.73*	26.02*	40.83*	31.78
TTL-160m	26.15	29.29	28.11	41.12	31.16
Pythia-160m	24.06*	31.39*	24.86*	44.34*	31.16
OPT-125m	22.87*	31.47*	26.02*	42.87*	30.80
GPorTuguese-2	22.48	29.62	27.36	41.44	30.22
Gpt2-small	21.48*	31.60*	25.79*	40.65*	29.97
Multilingual GPT	23.81	26.37*	25.17*	39.62	28.73

Table 4: All evaluations used the Language Model Evaluation Harness standard settings. Given our constrained budget, we could only evaluate some models in the Brazilian Portuguese version of the used benchmarks. Unfortunately, HellaSwag and the MMLU benchmarks require considerable time to run, even more so as the model size increases. Hence, results marked with an "*" were extracted from the Open LLM Leaderboard (Beeching et al., 2023), which uses the same evaluation method, using their English version. Thus, these results may vary if conducted in different languages, especially given that some models were pre-trained on mainly English text. Regardless, these results show that our models can perform as well and, in some instances (ARC and MMLU), surpass models trained on a much larger dataset and with many more resources. We speculate that this may result from our mixed dataset, which contains many demonstrations of instruction following and Q&A.

Figure 2: TTL-460 accuracy on the ARC-Challenge during training
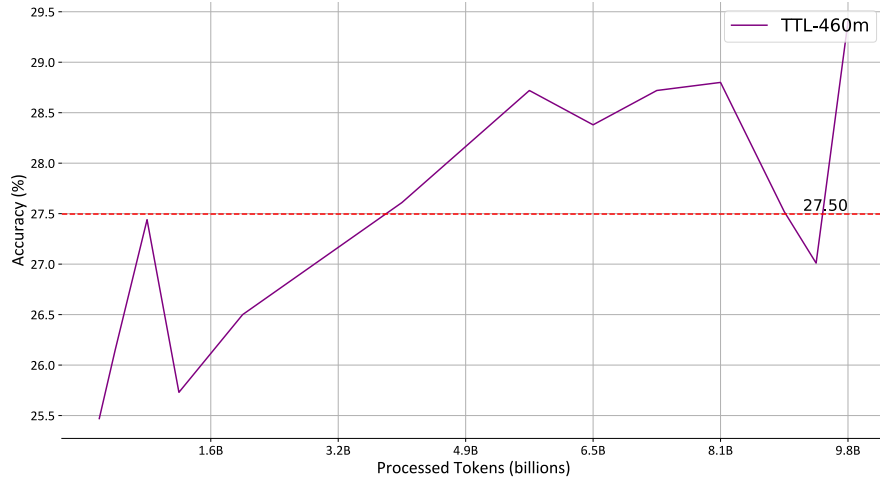

Figure 2: According to the average accuracy score of TTL-460m on the ARC-Challenge, our model outperforms all other models listed in Table 4. Given our restricted budget, we could not evaluate all four benchmarks over our 68 checkpoints.

evaluations, we fine-tuned all models considered using the same setting and compared their final performance in Table 5.

Table 5: Downstream performance on different tasks

Models	IMDB	FaQuAD-NLI	HateBr	Assin2	AgNews	Average
BERTimbau-large	**93.58**	92.26	91.57	**88.97**	94.11	92.10
BERTimbau-small	92.22	**93.07**	91.28	87.45	94.19	91.64
TTL-460m	91.64	91.18	**92.28**	86.43	**94.42**	91.19
TTL-160m	91.14	90.00	90.71	85.78	94.05	90.34

Table 5: All the shown results are the higher accuracy scores achieved on the respective task test sets after fine-tuning the models on the training sets. All fine-tuning runs used the same hyperparameters: 3 epochs, batch size of 16, AdamW as the optimizer ($\alpha = 4e^{-5}$, $\varepsilon = 1e^{-8}$), and a weight decay rate of 0.01. Even though bidirectional encoder-only models usually perform better on the types of tasks under consideration, our larger model (TTL-410) can, in this simple fine-tuning setting, outperform BERTimbau-large on tasks involving toxicity detection and general text classification. Given that all results from our models present an over 90% average accuracy score across tasks (even without pushing for optional hyperparameter settings), we argue that this shows the potential for our language models to be performative in many types of downstream tasks. All fine-tuned versions of TTL and their respective fine-tuning scripts are available on Hugging Face.

We also measured the efficiency of our model in terms of its throughput capabilities. Given that generative language models in real-time applications are critically tied to their throughput and memory footprint, we estimated how many tokens our models can generate per second (t/s). According to our test, on average, TTL-460m can generate up to $12_{t/s}$ on a Tesla V4 GPU. Applying a 4-bit activation-aware weight quantization (Lin et al., 2023) increases the throughput to $25_{t/s}$, and reduces the model's memory footprint to 340 MB. On an A100, we increase throughput to $60_{t/s}$, approximating 80 words generated per second. Further improvements can be achieved using inference frameworks that utilize more high-performance languages, like C or C++ (e.g., Llama.cpp).

Lastly, we licensed all models created by this project under an Apache 2.0 License.

4.2 Energy Consumption and Carbon Emissions

Given the consensus that tracking energy consumption, estimating carbon emissions, and reporting these results should be standard practice in the field of deep learning (Strubell et al., 2019; García-Martín et al., 2019; Lottick et al., 2019; Lacoste et al., 2019; Desislavov et al., 2021; Luccioni et al., 2022), we logged our energy consumption during training and evaluation runs by using CodeCarbon (CodeCarbon, 2019). Besides achieving a count for the total energy consumption of our project, this allowed us to monitor performance and energy consumption increases coupled during a training run of an LLM. Table 6 shows the logs for TTL-460m.

According to the estimations proposed by Lottick et al. (2019), implemented in CodeCarbon, in total, the 36 hours of compute time to train TTL-160m consumed 15.5 kWh ($\approx$ 5.7 KgCO2eq), while the 280 hours used to train TTL-460m consumed 113.0 kWh ($\approx$ 41.3 KgCO2eq). Summing all up, these emissions equate to a 185-kilometer car ride.[7][8]

4.3 Alignment

With the release of ChatGPT in November 2022, there has been an increase in interest in models that went through an alignment process (e.g., instruction tuning, preference modeling, etc.), making them more attuned to follow the commands of people without the need for sophisticated prompting or further fine-tuning, becoming, in general, more helpful tools (a.k.a. assistants) to their users. Nowadays, there are many assistant models like ChatGPT (Corrêa, 2023a; Taori et al., 2023; Touvron et al., 2023b; Jiang et al., 2023; Geng et al., 2023; Conover et al., 2023; Köpf et al., 2023), which, besides being an object of interest to the general public, have become one of the most used laboratories for alignment research (Askell et al., 2021).

---

[7]Calculations were made using the region of North Rhine-Westphalia (Germany) as the region of computing.

[8]GPU stats indicate that our training runs kept a steady allocation of GPU memory utilization (between 70% - 85% of its maximum capacity), power usage (83%), and thermal output ($\approx$ 60° C).

Table 6: Energy consumption during training (TTL-460m)

Processed Tokens	Perplexity	Energy Consumption (kWh)	Emissions (KgCO2eq)
8.1M	20.49	9.40	3.34
1.6B	16.90	18.82	6.70
2.4B	15.43	28.59	10.16
3.2B	14.64	38.20	13.57
4.0B	14.08	48.04	17.07
4.9B	13.61	57.74	20.52
5.7B	13.25	67.32	23.92
6.5B	12.87	76.84	27.30
7.3B	12.57	86.40	30.70
8.1B	12.27	96.19	34.18
9.0B	11.96	106.06	37.70
9.8B	11.77	115.69	41.31

Table 6: Here, we display how the perplexity score of our model diminishes at every 100,000 steps (8.1 million tokens) and what the energy consumption (kWh) and estimated carbon emissions (CO2eq) are related to this process. As one can see, after half of our training run, our performance increase slows down as the rate of consumption and emissions keeps following a linear trend. This shows that, on our training runs, almost half of the energy we consumed was tied to a marginal increase in the model's performance (≈ 1.84). By analyzing the training loss logs of other models (Black et al., 2022; Touvron et al., 2023b; Zhang et al., 2024), , we argue that this is a common reality in the training of large-scale neural networks, i.e., *convergence is slow and costly.*

Our base models can follow instructions with minimal prompting, given that they were already exposed to millions of instructions during training. To further expand these capabilities, we fine-tuned the 460m parameter version of TTL on an instructional dataset to create a chat version of our larger base model, TTL-460m-Chat. Like in the Alpaca study (Taori et al., 2023), we trained our chat model via SFT on a synthetically generated dataset. This dataset contains a collection of single-turn conversations between an assistant and a user, generated by prompting models that already went through an alignment process (ChatGPT, Vicuna, LLama 2, Open-Assistant, etc.). The dataset is available in Brazilian Portuguese and English and contains approximately 81K samples (Corrêa, 2023b).[9]

For the SFT, we used the same software stack utilized to pre-train our models. TTL-460m-Chat was trained for three epochs using almost the same configurations documented in Table 3. The only modifications are in the number of warm-up steps (1,000) and learning rate (1×10^{-5}). The full details are available in our GitHub repository.[10] We also created an open demo of our Chat model, which allows users to have conversations with TTL-460m-Chat.[11] In the demo, we also implemented a simple vector search engine that allows users to explore the fine-tuning dataset quickly, allowing them to evaluate the model's capabilities regarding "how much the model can go beyond its fine-tuning distribution." We also made available a 4-bit quantized version of this model for faster inference with almost no loss in performance. Figure 3 shows a sample of TTL-460m-Chat capabilities in story generation.

5 Limitations

Our work documents the process of developing text-generation language models for low-resource languages while being under a condition of low resources. With a budget of 500,00 USD, there is much that we could not perform, like multi-GPU distributed training with larger batch sizes and prolonged training runs. Even though our results indicate that we can achieve reasonable performance by limiting a training run to the stipulations made by specific scaling laws (Hoffmann et al., 2022), regardless, that evidence points out that our models are still under-trained. However, aware of the limitations of our work, we logged and documented every step in a reproducible manner, allowing

---

[9]The Brazilian Portuguese version is a translated version of the English one, generated via the Google Translate API.

[10]The additional training TTL-460m-Chat equated to an energy consumption of 5.6 kWh and an emission of 2.5 KgCO2eq.

[11]Available in https://huggingface.co/spaces/nicholasKluge/TeenyTinyLlama-Chat.

Figure 3: Sample generated by TTL-460m-Chat

😀: Escreva uma história sobre uma menina chamada Sophia - a domadora de gnomos.

😈: Sophia sempre foi fascinada por animais e desde pequena adorava explorar o mundo ao seu redor. Ela passava horas brincando com seus amigos peludos, explorando as florestas e rios próximos e até mesmo montando em jipes para passear. Mas à medida que ela crescia, sua paixão só se intensificava.

Um dia, enquanto caminhava pela floresta, Sophia tropeçou em um pequeno riacho. Ao dar um passo à frente, sentiu uma onda repentina de energia percorrendo suas veias. De repente, ela percebeu que estava sendo puxada pelo ar!

A princípio, Sophia ficou assustada. Ela nunca tinha visto nada parecido antes. Mas quando olhou mais de perto, viu algo incrível: os gnomos estavam voando, cada um deles girando e dançando no ar. E então, tão rapidamente quanto apareceu, eles desapareceram.

Sophia correu freneticamente pelas árvores, tentando encontrar o caminho de volta para casa. Mas não havia como voltar atrás. Os gnomos continuaram a voar e, eventualmente, chegaram a uma clareira onde Sophia finalmente encontrou o caminho de volta para casa.

Enquanto observava os gnomos desaparecerem, Sophia sabia que havia encontrado sua verdadeira paixão. Ela passou inúmeras noites estudando os animais, aprendendo tudo o que pôde sobre eles e os incríveis poderes que possuíam. E embora tenha sido difícil, ela também descobriu algo muito valioso – que às vezes as coisas mais bonitas da vida são aquelas que nos deixam sem fôlego e inspiradas.
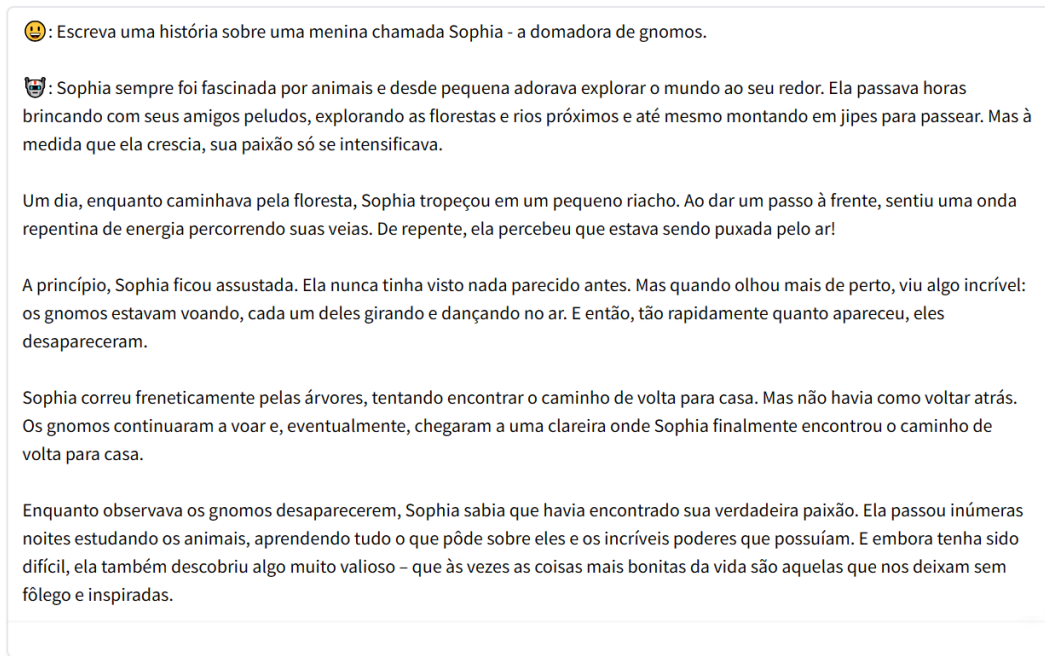
Figure 3: Above is a user prompt requesting the model to generate a story; the English translation equates to: "*Write a story about a girl called Sophia - the gnome tamer.*" Model Output: "*Sophia has always been fascinated by animals, and from an early age, she loved exploring the world around her. She would spend hours playing with her furry friends, exploring the nearby forests and rivers, and even riding in jeeps for rides. But as she grew older, her passion only intensified. One day, while walking through the forest, Sophia stumbled across a small stream. As she stepped forward, she felt a sudden surge of energy coursing through her veins. Suddenly, she realized that she was being pulled through the air! At first, Sophia was frightened. She had never seen anything like it before. But when she looked closer, she saw something incredible: the gnomes were flying, each spinning and dancing in the air. And then, as quickly as they had appeared, they disappeared. Sophia ran frantically through the trees, trying to find her way back home. But there was no turning back. The gnomes continued to fly and eventually reached a clearing, where Sophia finally found her way home. As she watched the gnomes disappear, Sophia knew she had found her true passion. She spent countless nights studying the animals, learning everything about them and their incredible powers. And although it was difficult, she also discovered something precious - that sometimes the most beautiful things in life leave you breathless and inspired.*"

others to push our work beyond what we currently can or begin new projects from our source. We still intend to run many tests and look forward to continuing to update this project.

Another aspect that limits this and every other work in language model development for low-resource languages is the need for more standard benchmarks to test the yields of our work. Machine learning is, to a great extent, a field moved by benchmarks. These tests allow us to define reasonable objectives and goals, work under a united evaluation framework, and compare our results in a standardized form. With this, progress becomes more accessible, and in this work, we tried to use evaluation methods that followed this motto. However, these are still limited, and there is a need to expand evaluation methodologies to encompass more languages. The field requires more work to translate or adapt current benchmarks to multiple languages. On the contrary, much of the research done in low-resource languages will continue to be (1) hard to reproduce and (2) difficult to compare.

Other limitations of this work are related to the TTL pair. Like almost all other language models trained on large text datasets scraped from the web, the TTL pair exhibited behavior that does not make them an out-of-the-box solution to many real-world applications, especially those requiring factual, reliable, nontoxic text generation. Our models are all subject to producing hallucinations, i.e., the generation of text that is incorrect, nonsensical, or not real, reproducing historical biases or generating toxic language, being overly verbose or repetitive, and generally producing content

that should not be taken as given without human moderation.[12] Hence, even though our models are released with a permissive license, we urge users to perform their risk analysis on these models if intending to use them for real-world applications and also have humans moderating the outputs of these models in applications where they will interact with an audience, guaranteeing users are always aware they are interacting with a language model.

6   Future Works

The utility of language models, combined with the fact that their capabilities, for most languages, are still under-explored, presents many opportunities for researchers. In this work, we sought to document and expose the challenges of picking these low-hanging fruits while sharing our methods and tools with the community.

Regarding the TTL pair and LLM development on a smaller scale in general, we believe there is much room to be explored regarding their use and utilities, given that a model tailored for low-resource scenarios can be used for many applications, like edge computing, game development, real-time applications, and more. Besides, these sorts of models are necessary artifacts for research in general. Having models trained natively in the language of non-English speakers opens the doors to a whole population of non-English speakers who might be interested in research involving LLMs, be that on NLP, AI ethics, AI alignment, or other tangential fields.

With this in mind, here are some possible avenues for future projects seeking to expand our work:

1. **Scaling to the 1B parameter range.** Accelerate permits a simple way to scale training on multiple GPUs, and at the 1B range, we are bound to encounter the emergence of improved capabilities (Gunasekar et al., 2023). Given that our datasets, according to scaling laws for data-constrained models (Muennighoff et al., 2023), are sufficient to train models up to that range, future training runs on multiple GPUs might give us the first billion parameter text generation models trained on Brazilian Portuguese text.

2. **Scaling dataset size to the 1T tokens mark.** To our knowledge, there are still no available trillion-sized datasets for Brazilian Portuguese. Collecting text datasets to create such a corpus would enable us to push the training of models to pass the Chinchilla scaling laws (Touvron et al., 2023a,b), explore the saturation and training limits of smaller language models (Zhang et al., 2024).

3. **Adding Brazilian Portuguese benchmarks to standard evaluation frameworks.** The more benchmarks we can bring to low-resource languages, the brighter future research for that language will be. Hence, adding tasks, new or translated versions, to evaluation frameworks like Gao et al. (2021) language model evaluation harness is the type of grassroots work bound to benefit an entire research community. Also, testing all the Brazilian Portuguese language models mentioned in this work under a united framework would produce insightful results.

4. **Expand the open-source development of language models for low-resource languages.** Our experiments could be reproduced for other languages if they can access a minimum amount of tokens. At the same time, open-source development should be encouraged (Köpf et al., 2023; Geng and Liu, 2023; Dey et al., 2023), so in the future, the whole concept of "*low-resource language*" will be a thing of the past.

---

[12]An interesting limitation of our models is how they generate erroneous code. Given that much of our training came from translated conversations on subjects like coding, our model tends to create code with programmatic commands that are natively written in English (`import torch`) on Brazilian Portuguese (`importar torch`), given the unstructured way in which people translated these samples, i.e., no distinguishment from source code and natural language docstrings and comments.

7 Conclusion

Large language models spearhead a paradigm shift in natural language processing, while widespread applications keep catalyzing these technologies to further push the field's boundaries. However, it is crucial to acknowledge that the advancements in LLMs have not been universally distributed across all languages and remain unevenly accessible, underscoring the ongoing challenge of achieving linguistic inclusivity openly and equitably.

In this study, our primary objective was to meticulously document the obstacles encountered and insights gained while training language models for low-resource languages, navigating the constraints imposed by a limited computational budget, a scarcity of available data, and a lack of standardized evaluation regimes.

Through this project, we successfully crafted a pair of language models, the *TeenyTinyLlama* pair, trained in Brazilian Portuguese to the optimal range defined by the Chinchilla scaling laws. Remarkably, our findings indicate that these models exhibit comparable performance to other language models of similar size in various linguistic tasks while still demonstrating signs of possible undertraining.

All models, datasets, and source code developed in this study have been released under a permissive license, fostering open access and encouraging collaborative research within the academic community.

Acknowledgments

References

Abadji, J., Suarez, P. O., Romary, L., and Sagot, B. (2022). Towards a cleaner document-oriented multilingual crawled corpus. *arXiv preprint arXiv:2201.06642*.

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. (2023). Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.

Alabi, J. O., Adelani, D. I., Mosbach, M., and Klakow, D. (2022). Adapting pre-trained language models to african languages via multilingual adaptive fine-tuning. *arXiv preprint arXiv:2204.06487*.

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Goffinet, É., Hesslow, D., Launay, J., Malartic, Q., et al. (2023). The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Antoun, W., Baly, F., and Hajj, H. (2021). AraGPT2: Pre-trained transformer for Arabic language generation. In Habash, N., Bouamor, H., Hajj, H., Magdy, W., Zaghouani, W., Bougares, F., Tomeh, N., Abu Farha, I., and Touileb, S., editors, *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., Jones, A., Joseph, N., Mann, B., DasSarma, N., et al. (2021). A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. (2022). Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., Sanseviero, O., Tunstall, L., and Wolf, T. (2023). Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., et al. (2024). Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. (2023). Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., et al. (2022). Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Bruno Henrique (2024a). Caramelo 7b. `https://huggingface.co/Bruno/Caramelo_7B`.

Bruno Henrique (2024b). Harpia-7b-guanacolora. `https://huggingface.co/Bruno/Harpia-7b-guanacoLora`.

Carlo Moro (2024). Instruct-ptbr-enus-11m. `https://huggingface.co/datasets/cnmoro/Instruct-PTBR-ENUS-11M`.

Chen, T., Xu, B., Zhang, C., and Guestrin, C. (2016). Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. (2022). Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. (2018). Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

CodeCarbon (2019). Codecarbon: Track emissions from compute and recommend ways to reduce their impact on the environment. `https://github.com/mlco2/codecarbon`.

Computer, T. (2023). Redpajama: an open dataset for training large language models. `https://github.com/togethercomputer/RedPajama-Data`.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Conover, M., Hayes, M., Mathur, A., Xie, J., Wan, J., Shah, S., Ghodsi, A., Wendell, P., Zaharia, M., and Xin, R. (2023). Free dolly: Introducing the world's first truly open instruction-tuned llm.

Corrêa, N. K. (2023a). Aira.

Corrêa, N. K. (2023b). Instruct-aira dataset version 2.0. `https://huggingface.co/datasets/nicholasKluge/instruct-aira-dataset-v2`.

Cui, Y., Yang, Z., and Yao, X. (2023). Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Dao, T. (2023). Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Desislavov, R., Martínez-Plumed, F., and Hernández-Orallo, J. (2021). Compute and energy consumption trends in deep learning inference. *arXiv preprint arXiv:2109.05472*.

Dettmers, T., Lewis, M., Shleifer, S., and Zettlemoyer, L. (2022). 8-bit optimizers via block-wise quantization. *9th International Conference on Learning Representations, ICLR*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dey, N., Gosal, G., Khachane, H., Marshall, W., Pathria, R., Tom, M., Hestness, J., et al. (2023). Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster. *arXiv preprint arXiv:2304.03208*.

Ding, N., Chen, Y., Xu, B., Qin, Y., Zheng, Z., Hu, S., Liu, Z., Sun, M., and Zhou, B. (2023). Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Eisenschlos, J. M., Ruder, S., Czapla, P., Kardas, M., Gugger, S., and Howard, J. (2019). Multifit: Efficient multi-lingual language model fine-tuning. *arXiv preprint arXiv:1909.04761*.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. (2020). The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Gao, L., Tow, J., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., McDonell, K., Muennighoff, N., et al. (2021). A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*.

Garcia, G. L., Paiola, P. H., Morelli, L. H., Candido, G., Júnior, A. C., Jodas, D. S., Afonso, L. C. S., Guilherme, I. R., Penteado, B. E., and Papa, J. P. (2024). Introducing bode: A fine-tuned large language model for portuguese prompt-based task.

García-Martín, E., Rodrigues, C. F., Riley, G., and Grahn, H. (2019). Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, 134:75–88.

Geng, X., Gudibande, A., Liu, H., Wallace, E., Abbeel, P., Levine, S., and Song, D. (2023). Koala: A dialogue model for academic research. Blog post.

Geng, X. and Liu, H. (2023). Openllama: An open reproduction of llama. https://github.com/openlm-research/open_llama.

Gu, A. and Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Gugger, S., Debut, L., Wolf, T., Schmid, P., Mueller, Z., Mangrulkar, S., Sun, M., and Bossan, B. (2022). Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate.

Guillou, P. (2020). Gportuguese-2 (portuguese gpt-2 small): a language model for portuguese text generation (and more nlp tasks...). https://huggingface.co/pierreguillou/gpt2-small-portuguese.

Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., Del Giorno, A., Gopi, S., Javaheripi, M., Kauffmann, P., de Rosa, G., Saarikivi, O., et al. (2023). Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., Gonzalez-Agirre, A., Armentano-Oller, C., Rodriguez-Penagos, C., and Villegas, M. (2021). Maria: Spanish language models. *arXiv preprint arXiv:2107.07253*.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. (2020). Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Huang, L., Le Bras, R., Bhagavatula, C., and Choi, Y. (2019). Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

HuggingFace (2019). Tokenizers: Fast state-of-the-art tokenizers optimized for research and production. https://github.com/huggingface/tokenizers.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. (2024). Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ko, H., Yang, K., Ryu, M., Choi, T., Yang, S., Park, S., et al. (2023). A technical report for polyglot-ko: Open-source large-scale korean language models. *arXiv preprint arXiv:2306.02254*.

Kocetkov, D., Li, R., Allal, L. B., Li, J., Mou, C., Ferrandis, C. M., Jernite, Y., Mitchell, M., Hughes, S., Wolf, T., et al. (2022). The stack: 3 tb of permissively licensed source code. *arXiv preprint arXiv:2211.15533*.

Köpf, A., Kilcher, Y., von Rütte, D., Anagnostidis, S., Tam, Z.-R., Stevens, K., Barhoum, A., Duc, N. M., Stanley, O., Nagyfi, R., et al. (2023). Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Lacoste, A., Luccioni, A., Schmidt, V., and Dandres, T. (2019). Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

Lai, V., Ngo, N. T., Veyseh, A. P. B., Dernoncourt, F., and Nguyen, T. H. (2023). Open multilingual llm evaluation leaderboard. https://huggingface.co/spaces/uonlp/open_multilingual_llm_leaderboard.

Lankford, S., Afli, H., and Way, A. (2023). adaptmllm: Fine-tuning multilingual language models on low-resource languages with integrated llm playgrounds. *Information*, 14(12):638.

Larcher, C., Piau, M., Finardi, P., Gengo, P., Esposito, P., and Caridá, V. (2023). Cabrita: closing the gap for foreign languages. *arXiv preprint arXiv:2308.11878*.

Laurençon, H., Saulnier, L., Wang, T., Akiki, C., Villanova del Moral, A., Le Scao, T., Von Werra, L., Mou, C., González Ponferrada, E., Nguyen, H., et al. (2022). The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. (2021). Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*.

Leonardo Souza (2024). Samba. https://huggingface.co/lrds-code/samba-1.1B.

Lhoest, Q., Villanova del Moral, A., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Le Scao, T., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A., and Wolf, T. (2021). Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Li, H., Koto, F., Wu, M., Aji, A. F., and Baldwin, T. (2023). Bactrian-x : A multilingual replicable instruction-following model with low-rank adaptation.

Lin, J., Tang, J., Tang, H., Yang, S., Dang, X., and Han, S. (2023). Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*.

Lin, S., Hilton, J., and Evans, O. (2021a). Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Lin, X. V., Mihaylov, T., Artetxe, M., Wang, T., Chen, S., Simig, D., Ott, M., Goyal, N., Bhosale, S., Du, J., et al. (2021b). Few-shot learning with multilingual language models. *arXiv preprint arXiv:2112.10668*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lottick, K., Susai, S., Friedler, S. A., and Wilson, J. P. (2019). Energy usage reports: Environmental awareness as part of algorithmic accountability. *arXiv preprint arXiv:1911.08354*.

Luccioni, A., Viguier, S., and Ligozat, A. (2022). Estimating the carbon footprint of bloom, a 176b parameter language model, doi: 10.48550. *arXiv preprint ARXIV.2211.02001*.

Luo, Y., Kong, Q., Xu, N., Cao, J., Hao, B., Qu, B., Chen, B., Zhu, C., Zhao, C., Zhang, D., et al. (2023). Yayi 2: Multilingual open-source large language models. *arXiv preprint arXiv:2312.14862*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Maicon Domingues (2023a). Canarim-7b (revision 08fdd2b). https://huggingface.co/dominguesm/canarim-7b.

Maicon Domingues (2023b). Canarim-instruct-ptbr-dataset (revision c2de751). https://huggingface.co/datasets/dominguesm/Canarim-Instruct-PTBR-Dataset.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., and Sagot, B. (2020). CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Muennighoff, N., Rush, A. M., Barak, B., Scao, T. L., Piktus, A., Tazi, N., Pyysalo, S., Wolf, T., and Raffel, C. (2023). Scaling data-constrained language models. *arXiv preprint arXiv:2305.16264*.

Nagoudi, E. M. B., Chen, W.-R., Abdul-Mageed, M., and Cavusogl, H. (2021). Indt5: a text-to-text transformer for 10 indigenous languages. *arXiv preprint arXiv:2104.07483*.

Nguyen, T., Nguyen, C. V., Lai, V. D., Man, H., Ngo, N. T., Dernoncourt, F., Rossi, R. A., and Nguyen, T. H. (2023a). Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages.

Nguyen, X.-P., Aljunied, S. M., Joty, S., and Bing, L. (2023b). Democratizing llms for low-resource languages by leveraging their english dominant abilities with linguistically-diverse prompts. *arXiv preprint arXiv:2306.11372*.

Nicolas de Camaret (2024). Cabra. https://huggingface.co/nicolasdec/Cabra.

NousResearch (2024). Nous-hermes-2-yi-34b.

Ortiz Su'arez, P. J., Romary, L., and Sagot, B. (2020). A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Ortiz Su'arez, P. J., Sagot, B., and Romary, L. (2019). Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut f"ur Deutsche Sprache.

Ott, M., Shleifer, S., Xu, M., Goyal, P., Duval, Q., and Caggiano, V. (2021). Fully sharded data parallel: faster ai training with fewer gpus.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Overwijk, A., Xiong, C., and Callan, J. (2022). Clueweb22: 10 billion web documents with rich information. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3360–3362.

Pablo Filetti Moreira (2024). Gpt4all-j prompt generations pt. https://huggingface.co/datasets/pablo-moreira/gpt4all-j-prompt-generations-pt.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K. K., et al. (2023). Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*.

Pires, R., Abonizio, H., Rogério, T., and Nogueira, R. (2023). Sabi\'a: Portuguese large language models. *arXiv preprint arXiv:2304.07880*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., et al. (2021). Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. (2020). Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Real, L., Fonseca, E., and Oliveira, H. G. (2020). The assin 2 shared task: a quick overview. In *International Conference on Computational Processing of the Portuguese Language*, pages 406–412. Springer.

Rodrigues, J., Gomes, L., Silva, J., Branco, A., Santos, R., Cardoso, H. L., and Osório, T. (2023). Advancing neural encoding of portuguese with transformer albertina pt. *arXiv preprint arXiv:2305.06721*.

Rodrigues, R. C. (2023). Faquad-nli: a benchmark for textual entailment. `https://huggingface.co/datasets/ruanchaves/faquad-nli`.

Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., Adi, Y., Liu, J., Remez, T., Rapin, J., et al. (2023). Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Scheible, R., Thomczyk, F., Tippmann, P., Jaravine, V., and Boeker, M. (2020). Gottbert: a pure german language model. *arXiv preprint arXiv:2012.02110*.

Shazeer, N. (2020). Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Shazeer, N. and Stern, M. (2018). Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Shen, S., Hou, L., Zhou, Y., Du, N., Longpre, S., Wei, J., Chung, H. W., Zoph, B., Fedus, W., Chen, X., et al. (2023). Mixture-of-experts meets instruction tuning: A winning combination for large language models. *arXiv preprint arXiv:2305.14705*.

Shliazhko, O., Fenogenova, A., Tikhonova, M., Mikhailov, V., Kozlova, A., and Shavrina, T. (2022). mgpt: Few-shot learners go multilingual. *arXiv preprint arXiv:2204.07580*.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. (2019). Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Souza, F., Nogueira, R., and Lotufo, R. (2020). Bertimbau: pretrained bert models for brazilian portuguese. In *Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part I 9*, pages 403–417. Springer.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*.

Su, J., Lu, Y., Pan, S., Wen, B., and Liu, Y. (2021). Roformer: Enhanced transformer with rotary position embedding. corr abs/2104.09864 (2021). *arXiv preprint arXiv:2104.09864*.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. (2023). Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Team, S. A. L. (2024). Stable lm 2 1.6b.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023a). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023b). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vargas, F., Carvalho, I., de Góes, F. R., Pardo, T., and Benevenuto, F. (2022). Hatebr: A large expert annotated corpus of brazilian instagram comments for offensive language and hate speech detection. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7174–7183.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wagner Filho, J. A., Wilkens, R., Idiart, M., and Villavicencio, A. (2018). The brwac corpus: A new open resource for brazilian portuguese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Wang, B. and Komatsuzaki, A. (2021). GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model.

Weights&Biases (2017). Weights & biases: A tool for visualizing and tracking your machine learning experiments. https://github.com/wandb/wandb.

Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., and Grave, E. (2020). CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Wikimedia Foundation (2024). Wikimedia Downloads. https://dumps.wikimedia.org.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Workshop, B., Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., et al. (2022). Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Xue, F., Fu, Y., Zhou, W., Zheng, Z., and You, Y. (2023). To repeat or not to repeat: Insights from scaling llm under token-crisis. *arXiv preprint arXiv:2305.13230*.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2020). mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. (2019). Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Zhang, B. and Sennrich, R. (2019). Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.

Zhang, P., Zeng, G., Wang, T., and Lu, W. (2024). Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. (2022). Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhang, X., Zhao, J. J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *NIPS*.

Zhao, J., Zhang, Z., Zhang, Q., Gui, T., and Huang, X. (2024). Llama beyond english: An empirical study on language capability transfer. *arXiv preprint arXiv:2401.01055*.