Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 267 (2025) 146–156

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Third EuroHPC user day

Pleias 1.0: the First Ever Family of Language Models Trained on Fully Open Data

Pierre-Carl Langlais[a,*], Pavel Chizhov[a,b], Mattia Nee[a], Carlos Rosas Hinostroza[a], Matthieu Delsart[a], Irène Girard[a], Anastasia Stasenko[a], Ivan P. Yamshchikov[a,b]

[a] *PleIAs, Paris, France*
[b] *THWS, Würzburg, Germany*

Abstract

Linguistic diversity and strong generalization in foundation language models are typically achieved by training on trillions of data tokens with very large model parameter counts. However, most such training datasets include substantial amounts of copyright-protected or private data that is not explicitly published under the licence that is permissive for LLM training, raising legal and ethical concerns. We introduce **Pleias 1.0**, a family of comparatively small foundation language models (with at most 3 billion parameters) trained *exclusively on public domain or permissively licensed data*. We release the model weights and training code so that our results are fully auditable and reproducible. Furthermore, we fine-tune our models for the Retrieval-Augmented Generation (RAG) task and demonstrate that these models – despite their smaller size – can outperform competitors that have orders of magnitude more parameters on RAG evaluations. All models, data, and code are released under open licenses, offering a new standard for transparency and compliance.

Keywords: fully open source language models; multilingual small language models;

1. Introduction

Large language models (LLMs) are typically trained on massive web-scraped corpora comprising hundreds of billions to trillions of tokens [8]. Prominent datasets such as C4, RefinedWeb, and Dolma consist largely of general web text, which often contains significant portions of copyrighted content and personal information [14, 15]. This practice raises legal and ethical issues, especially under emerging regulations (e.g., the EU AI Act) that demand transparency in data provenance and respect for privacy. Studies have also highlighted quality problems in web-derived data, including the presence of noise and toxic content [7], as well as inconsistencies and lower quality in non-English

* Corresponding author: Pierre-Carl Langlais.
E-mail address: pierre-carl@pleias.fr

1877-0509
10.1016/j.procs.2025.08.241

text [16, 21]. Moreover, the availability of web data for training is diminishing as major platforms containing potential training data restrict scraping and reuse.

These challenges have spurred interest in developing LLMs using only legally unencumbered data. Early efforts like the Pile dataset assembled a large collection of public text sources for training LLMs (825 GiB, 22 datasets) [8]. More recently, the AI community has launched projects to build models on transparent and open data (e.g., the Semantic Scholar Open Research Corpus [14] and the RedPajama training corpus [26]. However, to date, no language model of competitive performance has been trained using *only* data that is entirely in the public domain or under a permissive license. This work fills this gap.

We present **Pleias 1.0**, the first family of language models trained exclusively on open data at scale. Pleias 1.0 models set a new standard for legal compliance and transparency: all training data comes from the public domain or permissively licensed sources, making these models compliant by design with data provenance requirements and mitigating copyright concerns. Our models are relatively small in size (350M, 1.2B, and 3B parameters) yet are *multilingual*, adhering to instruction in eight European languages, and *high-performing on specialized tasks*. In particular, we target applications in retrieval-augmented generation (RAG), where a language model must accurately incorporate retrieved evidence into its outputs. We fine-tuned two of the Pleias models for RAG and found that they outperform other open models with up to 10× more parameters on our evaluation benchmarks.

Pleias 1.0 models are trained on the second core contribution presented in this work – **Common Corpus**[1], a new open multilingual dataset totaling approximately 2 trillion tokens. This corpus draws on diverse domains such as literature and historical text, government and legal documents, scientific publications, software code, and more. We developed a suite of data processing tools to maximize quality, including specialized models to correct OCR artifacts and filters to remove toxic or low-quality content. The resulting dataset provides broad linguistic coverage and high-quality text suitable for training state-of-the-art LLMs without relying on any proprietary web crawl data.

We train Pleias models from scratch on Common Corpus using an efficient distributed training pipeline built entirely with open-source tools. Despite their modest size, the Pleias models demonstrate strong capabilities. Notably, they exhibit best-in-class language coverage and adherence among models in the sub-1B scale: for example, our 350M model (Pleias Pico) fluently supports at least eight languages (English, French, Spanish, German, Italian, Dutch, Latin, Portuguese) and is less prone to unintended language switching, thanks to the balance of multiple languages in the data and a custom tokenizer design [5]. We also report that our training process was energy-efficient, yielding a significantly lower carbon footprint compared to other models of similar scale.

In summary, our contributions include:

- the creation of Common Corpus, a 2-trillion-token open multilingual dataset for LLM pretraining;
- the development of the Pleias 1.0 family of LLMs (350M, 1.2B, 3B parameters) trained entirely on this open corpus;
- novel training techniques and data processing pipelines to ensure data quality and efficient training;
- fine-tuning and evaluation showing that small, specialized models can match or beat much larger models on RAG tasks;
- the open release of all models, data, and code under permissive licenses to foster transparent research and practical adoption.

We hope this work paves the way for more ethical and accessible large language models. In the next section, we describe Pleias 1.0 pre-training dataset – Common Corpus – that we have collected, documented, and published for further use by the community. Section 3 describes the architecture of Pleias 1.0 and provides details on the training, including the carbon footprint of the models. Section 4 provides evaluations of the models. Section 5 is dedicated to the discussion of the difficulties we encounter and possible directions for further work.

[1] https://huggingface.co/datasets/PleIAs/common_corpus

2. Common Corpus: An Open Multilingual Dataset

2.1. Data Sources and Composition

Common Corpus is a new large-scale dataset composed entirely of text from sources that are either in the public domain or released under an open license. The corpus spans a wide range of domains and languages, reflecting our goal of broad coverage for a European multilingual foundation model. Major components of Common Corpus include:

- **Cultural Heritage Texts:** A collection of literature and historical texts that are out of copyright, including books, newspapers, and manuscripts from digital libraries and archives. For example, we incorporate texts from Project Gutenberg and national libraries (where copyright has expired) to represent classic literature and historical documents.
- **Legal and Government Documents:** A large body of legal texts and administrative documents released as open data. This includes legislative texts, court decisions, and government publications from jurisdictions that offer these under permissive licenses. For instance, we include portions of the EU's EuroParl and EurLex corpora (transcriptions of parliamentary sessions and legal directives) [17], and other public domain legal resources.
- **Scientific Publications:** Research papers and academic content available under open access terms. We draw on sources such as arXiv and the Semantic Scholar Open Research Corpus [14] for scientific and scholarly text in multiple languages. This component ensures the corpus contains high-quality formal writing and technical content.
- **Open Web Content:** Web text that is explicitly under open licenses. Unlike many LLM training sets that scrape random web pages, we restrict this category to permissively licensed sites and contributions. The largest portion here is Wikipedia (all languages), which is available under the CC BY-SA license. We also include other wiki-style or collaboratively created resources, and a filtered subset of Common Crawl data focusing only on documents with clear open licenses (e.g., certain government or news websites with open terms).
- **Code and Software Documentation:** Source code and software documentation released under open-source licenses. For this, we incorporated datasets like CodeSearchNet [11] (which collects code from GitHub under permissive licenses) and a filtered dump of Stack Exchange (which is CC BY-SA licensed Q&A content, including many programming discussions). This component introduces multilingual programming languages and technical problem-solving texts.
- **Other Niche Domains:** We also included various smaller open datasets to increase diversity, such as public domain subtitles, transcriptions of public speeches, ancient language texts (e.g., Latin corpora), and so on. All data sources were vetted to ensure compliance with our open license criteria.

Table 1 summarizes the language distribution of Common Corpus. The dataset is predominantly English (due to the abundance of English open content), but about 35–40% of the tokens are in languages other than English, making it one of the most multilingual open corpora of this scale. Notably, major European languages like French and German each contribute substantial portions, and the corpus includes content in dozens of languages overall. We prioritized including as many high-quality non-English sources as possible to improve the model's multilingual capabilities.

Table 1. Top 10 languages in Common Corpus by proportion of tokens.

Language	Percent of Tokens	Language	Percent of Tokens
English	64.4%	Dutch	1.45%
French	14.8%	Italian	1.26%
German	6.68%	Polish	0.67%
Spanish	2.62%	Greek	0.64%
Latin	2.03%	Portuguese	0.53%

Since Common Corpus is not the core focus of this paper, we refer the reader to the corresponding technical report for more details[2].

2.2. Data Processing and Quality Filtering

All data in Common Corpus undergoes an extensive processing pipeline to maximize quality and cleanliness. We develop several custom tools as part of this pipeline:

- **OCR Correction:** A significant portion of our cultural heritage data and some government documents come from scanned books or PDFs, which contain OCR (Optical Character Recognition) errors (e.g., incorrect characters, broken text segments). We trained a specialized language model called *OCRonos*[3] to automatically correct such digitization artifacts. OCRonos can identify common OCR issues like character substitutions, word splits/merges, and formatting errors, and output a cleaned text that remains faithful to the original. By applying OCRonos to all texts that were derived from image scans, we dramatically improved the readability and accuracy of these portions of the corpus. An important aspect is that OCRonos is multilingual and was itself trained on a mix of diverse OCR-ed texts, so it generalizes well across languages.
- **Text Segmentation:** For certain document collections (particularly historical archives), raw text extractions can be poorly segmented (e.g., multiple articles concatenated, or text order scrambled). We utilized another internal tool, *Segmentext*[4], a specialized model for text segmentation. Segmentext is designed to handle broken or unstructured text and determine proper segment boundaries even in the presence of noise. This helped in splitting long concatenated texts into coherent segments (e.g., separating individual articles in a newspaper scan batch).
- **Quality Filtering:** We applied systematic filtering to remove low-quality content. This included rule-based and model-based filters to eliminate gibberish or extremely repetitive texts, as well as to down-sample overly similar or duplicated content. We also filtered out content that was too short or too long, since extremely short texts are not very useful for language modeling, and extremely long texts (exceeding our context length significantly) might pose difficulties in training.
- **Toxicity and Bias Mitigation:** Although our data sources are generally cleaner than an uncurated web crawl, there still exists the possibility of toxic language, hate speech, or other undesirable content in a corpus of this size (for example, historical texts can contain archaic prejudiced language). To address this, we integrated a *toxicity classifier* in the pipeline to identify and remove highly toxic or hateful segments. This classifier was developed with a combination of keyword heuristics and a pre-trained model for toxicity detection. By filtering out flagged content, we aimed to reduce the incidence of harmful outputs from the trained model. We report further details about toxicity classification and filtering in a separate work [3].
- **Synthetic Data Augmentation:** In addition to real-world text, we augmented Common Corpus with a relatively small portion of synthetic text (approximately 30 billion tokens, ~1.5% of the total). This synthetic data was generated to provide examples of tasks or domains underrepresented in the corpus (such as conversational QA format, which can help with downstream fine-tuning). However, we were cautious to limit the amount of synthetic data. Recent studies have observed that excessive use of synthetic data can lead to performance plateaus or degradation in LLM training. In fact, Dohmatob et al. [6] show that mixing even a modest percentage of real data with predominantly synthetic data is crucial to maintain the benefits predicted by scaling laws. Guided by these findings, we only supplement with a small synthetic set to avoid dominating the training distribution.

After processing and filtering, the final Common Corpus is stored in a tokenized binary format for efficient reading during model training (see Section 3.2). Overall, our data pipeline significantly improves the signal-to-noise ratio of the corpus. We emphasize that while using only public data addresses legal transparency, it does not automatically

[2] https://huggingface.co/datasets/PleIAs/common_corpus
[3] https://huggingface.co/PleIAs/OCRonos
[4] https://huggingface.co/PleIAs/Segmentext

guarantee ethical perfection of the data (e.g., public domain texts can still contain biases or outdated viewpoints). Nonetheless, by carefully curating sources and filtering, we aim to produce a dataset that is not only legally clean but also of high quality for modeling.

3. Model Architecture and Training

3.1. Model Design

We train three model variants as parts of the Pleias 1.0 family, corresponding to parameter counts of roughly 350 million, 1.2 billion, and 3 billion. All are decoder-only Transformer language models with architectures inspired by recent open LLMs like LLaMA [24] and GPT-NeoX [4]. Table 2 summarizes the key architecture hyperparameters for each model size.

Table 2. Pleias 1.0 model architecture configurations. All models use decoder-only Transformer layers with SwiGLU activation and rotary positional embeddings.

Model	#Layers	Model Dim	FFN Dim	#Heads	KV Heads	Context Length
Pleias 1.0 350M	26	1024	2560	16	8	2048
Pleias 1.0 1.2B	22	2048	6144	32	8	2048
Pleias 1.0 3B	22	3072	12288	32	8	4096

All Pleias models use pre-normalization (LayerNorm at the start of each block) and employ the SwiGLU gated activation unit in feed-forward layers, following Shazeer [20]. We use Rotary Positional Embeddings (RoPE) [22] for encoding position within the self-attention mechanism, with a base period of 10,000. RoPE allows extrapolation to longer contexts, which we leverage by training the 3B model with a 4096 token context window (doubling the standard 2048 of the smaller models).

3.2. Tokenizer

We build a custom Byte-Pair Encoding (BPE) tokenizer for the Pleias models to accommodate the multilingual nature of Common Corpus. We train a tokenizer with 65,536 unique tokens on a representative sample of the corpus. We also use special tokens for padding, unknown words, and text beginnings and endings. Unlike some standard tokenizer configurations (e.g., the original LLaMA tokenizer), we remove certain English-centric preprocessing rules, such as those that split on specific Unicode punctuation or contract apostrophes, to treat all languages more uniformly. The resulting tokenizer provides more equitable subword segmentations across languages, ensuring that, for example, French or German text is not tokenized into disproportionately longer sequences than English.

We find that using a fresh tokenizer was preferable to reusing an existing one from a model like GPT-2 or LLaMA, because recycling tokenizers can introduce hidden biases or inefficiencies for languages not well-represented in the original tokenizer's training data [2]. By training on our corpus, the Pleias tokenizer achieves an average compression ratio (bytes to tokens) that is similar across our top languages.

For the RAG fine-tuned versions of Pleias (described in Section 3.5), we extend the tokenizer with a handful of special tokens used to delimit parts of the RAG prompt/response format. Specifically, tokens marking the start and end of a user query, the start and end of each retrieved source text, the beginning of the model's answer, etc., were added. These tokens enable the model to be aware of the structure in a RAG setting (e.g., keeping track of which text is a source and which is the answer) without ambiguity.

3.3. Training Setup and Hyperparameters

We train each model from scratch on the Common Corpus using an end-to-end open-source toolchain. Training was executed with the HuggingFace *Nanotron* library[5]. We first convert the entire corpus into packed tokenized sequences

[5] https://github.com/huggingface/nanotron

using the *Nanoset* data preparation tool[6]. This allows us to stream pre-tokenized batches during training, avoiding on-the-fly tokenization overhead.

For optimization, we use AdamW, and each training run used a linear warmup followed by a cosine decay learning rate schedule. The warmup phase lasts for 5–8% of total training steps (we used a slightly longer warmup for the larger models, based on early experiments). Table 3 details the training hyperparameters and parallelism setup for each model.

Table 3. Training configuration for Pleias 1.0 models.

Model	DP	TP	PP	Micro-batch	Gradient Accumulation	Total Batch Size	Peak LR → End LR
Pleias Pico (350M)	64	1	1	8	2	~2M	$3 \times 10^{-3} \rightarrow 3 \times 10^{-5}$
Pleias Nano (1.2B)	192	1	1	8	2	~6M	$1 \times 10^{-3} \rightarrow 1 \times 10^{-5}$
Pleias-3B Base	48	4	1	8	8	~12M	$5 \times 10^{-4} \rightarrow 5 \times 10^{-6}$

We perform training on two computing infrastructures. The 350M and 3B models are trained on the Jean Zay supercomputer in France (using H100 GPUs, under a Grand Challenge compute grant), while the 1.2B model was trained using *TractoAI*'s cloud platform[7]. In the latter case, we adapt our pipeline to TractoAI's serverless infrastructure backed by YTsaurus. This involved converting the pre-tokenized data into TractoAI table format and modifying Nanotron to read from this distributed storage, as well as custom handling of checkpoint saves. These engineering adaptations enable fault-tolerant, large-scale training on the cloud without proprietary software.

Each model is trained for a number of epochs on Common Corpus, with a curriculum between a "base" portion and a further "clean" portion of data. Specifically, for the 3B model, we trained for 2 epochs on the full Common Corpus (approximately 2 trillion tokens total) and then 1 additional epoch on a more aggressively filtered subset of the corpus (about 1 trillion tokens of the highest quality segments). This two-stage training (which we term an *annealing corpus*) is intended to first expose the model to very diverse data, then fine-tune it on cleaner data for the final phase. For the 1.2B model, we perform 1 epoch on the full corpus and 2 epochs on the clean subset. For the smallest 350M model, we find it sufficient to train on just the clean subset for 1 epoch (given computational constraints and the model's capacity). In total tokens seen, the 1.2B and 3B models each process on the order of 3–4 trillion tokens when counting multiple epochs, which is a substantial training budget for models of this size.

3.4. Carbon Footprint and Efficiency

One motivation for developing smaller, targeted models is to reduce the environmental impact of training and deploying LLMs. We calculate the carbon emissions from training our models using an online emissions calculator [12] that accounts for the energy draw of the GPU hardware and the carbon intensity of the power grid. Table 4 reports the training compute used and estimated emissions for each Pleias model, and compares them to other open models of similar scale.

Table 4. Training compute and estimated carbon emissions for Pleias 1.0 models, compared to other models. OpenELM refers to a model baseline of comparable size, and LLaMA refers to the hypothetical LLaMA training emissions scaled to a similar parameter count (based on publicly reported values). Emissions are in tonnes of CO_2 equivalent (tCO_2eq).

Model Size	#GPUs (type)	Time (days)	Pleias	Similar OpenELM	Similar LLaMA
~350M	64 (H100)	1.92	0.5 tCO_2	1.5 tCO_2	–
~1.2B	192 (H100)	5.0	4.0 tCO_2	5.5 tCO_2	107 tCO_2
~3B	192 (H100)	20.0	16 tCO_2	7.0 tCO_2	133 tCO_2

We observe that Pleias 1.0 models required an order of magnitude less CO_2 emissions to train compared to larger models like Meta's LLaMA (which in 7B-13B scale has been estimated at tens to hundreds of tons). Even against contemporaries in the small-model regime, Pleias models are very efficient. For instance, our 1.2B model consumed

[6] https://github.com/huggingface/nanotron/blob/main/docs/nanoset.md
[7] https://tracto.ai/

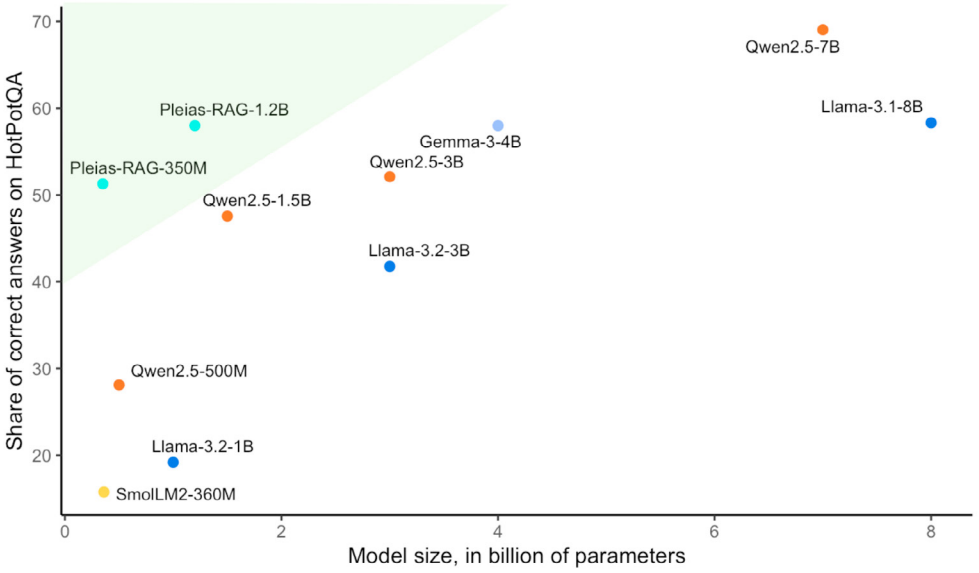

Fig. 1. Loss of RAG-fine-tuned Pleias 1.0 on non-English on HotPotQA [27]. Pleias 1.0 is in the Pareto-optimal part of the plot. For further details on the RAG-fine-tuned Pleias 1.0 see technical report [13].

about 4 tCO_2, which is about 30% lower than a comparable 1B-scale open model reported by Mehta et al. [18]. The efficiency is achieved through a combination of factors: smaller model size, a well-optimized training pipeline (high GPU utilization, large batch sizes), and the use of the hardware of the latest generation (NVIDIA H100), which offers better performance per watt.

The relatively low carbon footprint means that reproducing or fine-tuning these models is within reach of academic labs or smaller organizations that might have sustainability constraints. By releasing our training pipeline and configurations, we encourage others to build on this work in an energy-conscious way.

4. Evaluation

We evaluate the Pleias 1.0 models on both general language tasks and specialized retrieval-augmented generation tasks. Our focus is on demonstrating the capabilities of the RAG-tuned models relative to other open models of similar or larger scale. In addition, we provide some qualitative observations on eight European languages.

4.1. Fine-Tuning for Retrieval-Augmented Generation and RAG Evaluations

After pretraining the base models, we fine-tune two of them (350M and 1.2B) for retrieval-augmented generation, creating **Pleias 350M RAG** and **Pleias 1.2B RAG**, respectively. This fine-tuning is effectively an additional phase of language modeling training on a specialized corpus, rather than a typical small-sample supervised fine-tune. We construct a synthetic RAG training corpus of about 10 billion tokens comprising queries, retrieved passages, and annotated answers in separate regimes for trivial questions and questions that required structured reasoning. The content for this is derived from our Common Corpus knowledge sources (so as to remain in the realm of open data). The fine-tuning procedure continues training from the final checkpoint of the base model, using the same optimizer settings but at a lower learning rate (we reused the last learning rate from pretraining as the constant rate during RAG tuning). We detail the work on the RAG-fine-tuned models in the corresponding technical report [13].

The evaluations are performed on three question answering benchmarks: HotpotQA [27], 2WikiMultiHopQA [10], and MuSiQue [25], and compared them to the instruction-tuned models from LLaMA 3.1 [9], Qwen 2.5 [19], SmolLM 2 [1], and Gemma 3 [23] model families. To summarize the results, our models performed best compared to the

models of their size, and sometimes outperformed way larger ones. For example, PleIAs-RAG-350M outperformed all the tested counterparts, including the Qwen 2.5 7B, a model 20 times larger.

To highlight the multilingual capabilities of our models, we also translate the benchmarks to the main European languages. When evaluating the same set of models on the translated questions, we found that the performance of Pleias-RAG models remains high, with less than 5% of drop for the 1.2B model, while for the competitors, the general drop in performance ranges from 10% for Qwen 2.5 7B to more than 30% for SmolLM2 360M. This suggests that our models are more reliable in a multilingual setting, especially for their size range. This advantage can be largely attributed to the rich multilingual pre-training.

4.2. Multilingual Capabilities

Beyond English, the Pleias models demonstrate strong multilingual capabilities for the size of the models. Thanks to Common Corpus, which contains a rich variety of languages, our models can understand and generate several languages with high competency. As noted earlier, Pleias 1.0 350M is the first model of its size to "fully support" a range of major languages (at least eight). In practical terms, this means it can answer questions or hold a conversation in those languages without defaulting back to English or producing incoherent text.

We perform some supplementary tests to verify language adherence. For example, we asked each model a question in French and observed whether the answer stayed in French or switched to English. Pleias 350M and Pleias 1.2B reliably answered in French, whereas some baseline models like LLaMA-2 7B (chat) often responded in English to a French query, indicating a bias toward English. This adherence is important for a user experience perspective: a multilingual model should ideally reply in the language of the query unless instructed otherwise.

The custom tokenizer and balanced training data likely contribute to this behavior. Since our tokenizer does not over-fragment non-English text, the models do not see non-English inputs as overly rare or token-intensive. Combined with the fact that 35% of training tokens were non-English (with some languages like French making up nearly 15%), the models had sufficient exposure to learn those languages well. We also note that including Latin in the top languages is unusual for LLMs, which, however, is the case for Common Corpus because many public domain books and documents from history are in Latin; Pleias models gained some ability to read and write basic Latin, which might be beneficial for niche scholarly applications.

5. Encountered Problems

During the training of Pleias models, we encounter several problems that require changes and adaptations. In this section, we enumerate several of the most typical ones, which might help model developers in the future.

Cluster-framework compatibility. There are several open frameworks for distributed training of large models available, and their documentation usually does not include all the possible errors that can occur. The main sources of such errors are deep dependency libraries that are getting constantly updated and cluster setup difficulties that were not accounted for in the base documentation. To solve these problems, we consulted with the communities for the frameworks that were previously used in similar conditions and ran several tests with the main ones. In addition, for the smaller error, we used the search through StackOverflow[8] and LLM chatbots, mainly Claude[9].

Learning rate scheduling bug. When we run our training, Nanotron was a framework in the early stages of active development, so many new features were appearing, and together with them came the bugs. One of the bugs that affected our process was learning rate scheduling initialization, which was set up incorrectly when restarting from a checkpoint. Since we were doing regular audits of the training process, we were able to identify this issue early and submit it to the developers[10]. Since the framework was being rapidly developed and popular in the community, the issue was quickly identified, and we could integrate a solution to our local version of Nanotron before the actual integration of this fix into the framework.

[8] https://stackoverflow.com/questions
[9] https://claude.ai/
[10] https://github.com/huggingface/nanotron/issues/233

Cluster availability. We train the models on the Jean Zay supercomputer[11], which is used by plenty of other research groups, and, though having a considerable amount of allocated resources, might be overloaded. To overcome this difficulty, we started our training in advance and always chose the proper job timeout so that we do not clutter the Slurm[12] queue and do not get demoted by the scheduler. There are also two factors that allow us to predict higher cluster availability in the foreseeable future: right before and after the cluster maintenance, which happens quite regularly, and during the holiday periods.

Job timeout limit. To avoid long resource outages due to stagnating jobs, the Jean Zay cluster has an upper limit of 20 hours set for every job, so we cannot expect our jobs to run longer than this. To overcome this issue, we leveraged Slurm's arrayed jobs, which enabled us to set up a sequence of jobs running for the maximum allowed time. Close to the end of each job, we expected a checkpoint to be saved, so that the next job could start from this checkpoint. We also added a command to our Slurm scripts that would change the parameters of the current training run configuration, accounting for the job that is being started.

6. Conclusion

We introduce Pleias 1.0, a suite of small-scale language models trained entirely on open data, and demonstrate that these models can achieve outstanding results on specialized tasks like retrieval-augmented generation. By leveraging the Common Corpus – a massive, multilingual, public domain dataset – and carefully designing our training pipeline, we addressed key legal and ethical concerns (data transparency and licensing) without sacrificing performance. In fact, Pleias models match or surpass much larger models in our evaluations, all while being more efficient to train and run.

Our work underscores the importance of **data quality and domain-specific optimization**. Instead of blindly scaling model size or using indiscriminate web crawls, we focused on curating a training set that is legally clean and rich in the kinds of information our models need for high-value applications (like answering factual questions with evidence). The success of Pleias 1.2B and 350M in RAG tasks suggests that targeted smaller models can be viable alternatives to giant general-purpose models, especially for use cases where computational resources or latency are constraints.

We believe Pleias 1.0 sets a precedent for *fully open LLM development*: every component of our project – the data, the training code, the model weights, and even the evaluation data – is released openly. This enables anyone to reproduce or build upon our results, and provides a foundation for future research into compliant and transparent AI. Moreover, our approach is directly aligned with forthcoming regulatory requirements (such as the EU AI Act) that will likely necessitate knowing and disclosing exactly what data an AI model was trained on. Pleias 1.0 proves that such compliance is achievable today.

In future work, we plan to extend this paradigm in several directions. One is scaling up the model size while remaining within the open-data regime, to see if a 7B or 13B model trained on Common Corpus can further close the gap with the largest proprietary models. Another direction is expanding the multilingual coverage beyond mostly Indo-European languages, by incorporating more open data from diverse languages (e.g., expanding into Asian and African languages, where obtaining public domain text might require new collection efforts). We also aim to explore other task-specific fine-tunes of the base Pleias models, such as instruction tuning for general helpfulness or specialized domains like legal reasoning, again using only permissively licensed instruction data.

To the community, we release Pleias 1.0 models under the Apache 2.0 license, and the Common Corpus dataset under a CC BY license (with components in the public domain). We hope these resources will be useful for both researchers and practitioners. The 350M Pleias model, in particular, is light enough to run on consumer hardware, opening the door for truly local and private deployment of assistive LLMs that users can trust and verify. We publicly release the models and training data and invite the open science community to further contribute to creative applications of our models, as well as collaborate on the next iterations of open-data language models.

[11] http://www.idris.fr/eng/jean-zay/jean-zay-presentation-eng.html
[12] https://slurm.schedmd.com/documentation.html

Acknowledgements

The authors of this work acknowledge the HPC resource allocation by the European High Performance Computing Joint Undertaking (EuroHPC JU) on the Jean Zay cluster (compute grant #GC011015451).

References

[1] Allal, L.B., Lozhkov, A., Bakouch, E., Blázquez, G.M., Penedo, G., Tunstall, L., Marafioti, A., Kydlíček, H., Lajarín, A.P., Srivastav, V., Lochner, J., Fahlgren, C., Nguyen, X.S., Fourrier, C., Burtenshaw, B., Larcher, H., Zhao, H., Zakka, C., Morlon, M., Raffel, C., von Werra, L., Wolf, T., 2025. Smollm2: When smol goes big – data-centric training of a small language model. URL: https://arxiv.org/abs/2502.02737, arXiv:2502.02737.

[2] Arnett, C., 2023. Dangers of tokenizer recycling. Hugging Face Blog. https://huggingface.co/blog/catherinearnett/dangers-of-tokenizer-recycling.

[3] Arnett, C., Jones, E., Yamshchikov, I.P., Langlais, P.C., 2024. Toxicity of the commons: Curating open-source pre-training data. URL: https://arxiv.org/abs/2410.22587, arXiv:2410.22587.

[4] Black, S., et al., 2022. Gpt-neox-20b: An open-source autoregressive language model. arXiv preprint arXiv:2204.06745 .

[5] Chizhov, P., Arnett, C., Korotkova, E., Yamshchikov, I.P., 2024. BPE gets picky: Efficient vocabulary refinement during tokenizer training, in: Al-Onaizan, Y., Bansal, M., Chen, Y.N. (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA. pp. 16587–16604. URL: https://aclanthology.org/2024.emnlp-main.925/, doi:10.18653/v1/2024.emnlp-main.925.

[6] Dohmatob, E., Feng, Y., Kempe, J., 2024. Combating mode collapse in rlhf: On the connection between synthetic data and grokking in llms. arXiv preprint arXiv:2402.00159 .

[7] Elazar, Y., Bhagia, A., Magnusson, I., Ravichander, A., et al., 2023. What's in my big data? annotating and analyzing large web-crawled datasets. arXiv preprint arXiv:2308.00086 .

[8] Gao, L., Biderman, S., et al., 2021. The Pile: An 800gb dataset of diverse text for language modeling, in: Proceedings of NeurIPS Datasets and Benchmarks.

[9] Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C.C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., et al., 2024. The llama 3 herd of models. URL: https://arxiv.org/abs/2407.21783, arXiv:2407.21783.

[10] Ho, X., Duong Nguyen, A.K., Sugawara, S., Aizawa, A., 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps, in: Scott, D., Bel, N., Zong, C. (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online). pp. 6609–6625. URL: https://aclanthology.org/2020.coling-main.580/, doi:10.18653/v1/2020.coling-main.580.

[11] Husain, K., et al., 2019. Codesearchnet challenge: Evaluating the state of semantic code search. arXiv preprint arXiv:1909.09436 .

[12] Lacoste, A., Luccioni, S., Schmidt, T., Dandres, T., 2019. Quantifying the carbon emissions of machine learning. arXiv preprint arXiv:1910.09700 .

[13] Langlais, P.C., Chizhov, P., Nee, M., Hinostroza, C.R., Delsart, M., Girard, I., Hicheur, O., Stasenko, A., Yamshchikov, I.P., 2025. Even small reasoners should quote their sources: Introducing the pleias-rag model family. URL: https://arxiv.org/abs/2504.18225, arXiv:2504.18225.

[14] Lo, K., Wang, L.L., et al., 2020. S2ORC: The semantic scholar open research corpus, in: Proceedings of ACL.

[15] Longpre, S., Mahari, R., Muennighoff, N., Kabbara, J., Khazatsky, A., et al., 2023. The data provenance initiative: A large scale audit of dataset licensing & attribution in AI. arXiv preprint arXiv:2310.16787 .

[16] Luccioni, S., Viviano, J., 2021. What's in the box? an analysis of undesirable content in the common crawl corpus. arXiv preprint arXiv:2105.02732 .

[17] Martins, P.H., Fernandes, P., Alves, J., et al., 2024. EuroLLM: Multilingual language models for europe. arXiv preprint arXiv:2409.16235 .

[18] Mehta, S., Sekhavat, M., Cao, Q., Horton, M., Jin, Y., Sun, F., Mirzadeh, I., Najibikohnehshahri, M., Belenko, D., Zatloukal, P., Rastegari, M., 2024. Openelm: An efficient language model family with open training and inference framework, in: ICML Workshop. URL: https://arxiv.org/abs/2404.14619.

[19] Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z., 2025. Qwen2.5 technical report. URL: https://arxiv.org/abs/2412.15115, arXiv:2412.15115.

[20] Shazeer, N., 2020. Glu variants improve transformer. arXiv preprint arXiv:2002.05202 .

[21] Sikasote, C., Machovac, D., Kliegr, T., 2021. Quality at a glance: An audit of web-crawled multilingual datasets. arXiv preprint arXiv:2103.12028 .

[22] Su, J., Shen, T., Zhu, Z., et al., 2021. Roformer: Enhanced transformer with rotary position embedding, in: Advances in Neural Information Processing Systems (NeurIPS).

[23] Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., Rouillard, L., Mesnard, T., Cideron, G., bastien Grill, J., Ramos, S., Yvinec, E., Casbon, M., Pot, E., Penchev, I., Liu, G., Visin, F., Kenealy, K., Beyer, L., Zhai, X., Tsitsulin, A., Busa-Fekete, R., Feng, A., Sachdeva, N., Coleman, B., Gao, Y., Mustafa, B., Barr, I., Parisotto, E., Tian, D., Eyal, M., Cherry, C., Peter, J.T., Sinopalnikov, D., Bhupatiraju, S., Agarwal, R., Kazemi, M., Malkin, D., Kumar, R., Vilar, D., Brusilovsky, I., Luo, J., Steiner, A., Friesen, A., Sharma, A., Sharma, A., Gilady, A.M., Goedeckemeyer, A., Saade, A., Feng, A., Kolesnikov, A., Bendebury, A., Abdagic, A., Vadi, A., György, A., Pinto, A.S., Das, A., Bapna, A., et al., 2025. Gemma 3 technical report. URL: https://arxiv.org/abs/2503.19786, arXiv:2503.19786.

[24] Touvron, H., et al., 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 .

[25] Trivedi, H., Balasubramanian, N., Khot, T., Sabharwal, A., 2022. MuSiQue: Multihop questions via single-hop question composition. Transactions of the Association for Computational Linguistics 10, 539–554. URL: https://aclanthology.org/2022.tacl-1.31/, doi:10.1162/tacl_a_00475.

[26] Weber, M., Fu, D.Y., Anthony, Q., Oren, Y., Adams, S., Alexandrov, A., Lyu, X., Nguyen, H., Yao, X., Adams, V., Athiwaratkun, B., Chalamala, R., Chen, K., Ryabinin, M., Dao, T., Liang, P., Ré, C., Rish, I., Zhang, C., 2024. Redpajama: an open dataset for training large language models. NeurIPS Datasets and Benchmarks Track .

[27] Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R., Manning, C.D., 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. URL: https://arxiv.org/abs/1809.09600, arXiv:1809.09600.