What we Learned from Continually Training Minerva: a Case Study on Italian

Luca Moroni[1,*], Tommaso Bonomo[1], Luca Gioffré[1], Lu Xu[1], Domenico Fedele[2], Leonardo Colosi[2], Andrei Stefan Bejgu[2], Alessandro Scirè[2] and Roberto Navigli[1,2]

[1]Sapienza NLP Group, Dip. di Ingegneria Informatica, Automatica e Gestionale, Sapienza University of Rome, Rome, Italy
[2]Babelscape, Rome, Italy

Abstract

Modern Large Language Models (LLMs) are commonly trained through a multi-stage pipeline encompassing pretraining and supervised finetuning. While recent studies have extensively investigated the benefits of continual pretraining on high-quality data, these efforts have focused primarily on English. In this work, we explore the effectiveness of various data mixtures in a continual pretraining setting to enhance performance on Italian-language tasks. Leveraging Minerva-7B, a fully open-source LLM pretrained on a corpus composed of 50% Italian, we define and evaluate three distinct data recipes–comprising mathematical, encyclopedic, and copyrighted content–spanning both Italian and English. We also investigate the effect of extending the model's context window during continual pretraining on its ability to handle long-context tasks. To support our evaluation, we introduce INDAQA, a new benchmark for narrative question answering in Italian. Our results reveal that both data composition and increased context length substantially improve performance, offering valuable insights into continual pretraining strategies for less represented languages within an open scientific framework.

Keywords

Large Language Models, Italian, Continual Pre-training, Culturality, Long Context

1. Introduction

Modern Large Language Models (LLMs) are typically trained through a multi-stage process comprising pretraining, supervised fine-tuning (SFT), and preference alignment. During pretraining, models are trained in an autoregressive manner to learn language in an unsupervised way, without requiring human-labeled data [1, 2]. This phase allows models to acquire linguistic knowledge from large-scale, unstructured corpora. Recent approaches [3, 4, 5, 6] structure the pretraining process into two steps. In the first, models are exposed to trillions of raw web-sourced tokens, with only a small portion of high-quality content. In the second, training continues on a curated set of high-quality language or domain-specific texts, aiming to mitigate the impact of low-quality web content and extend the model's exposure to up-to-date and informative content.

After the intensive pretraining phase—where LLMs are trained solely on unlabeled data—models undergo supervised fine-tuning to adapt to real-world use cases. SFT can target either task-specific applications (e.g., question answering or summarization) or, more frequently, aim at training general-purpose conversational models. This is achieved by finetuning LLMs on hundreds of thousands of conversations covering diverse domains. Through this process, models learn to follow instructions to perform a wide range of tasks [7, 8, 9] and generate coherent responses in dialogue-like interactions.

While the overall LLM training pipeline has become increasingly standardized, the role of curated data after initial pretraining remains an active area of investigation for further improving model capabilities. However, the effects of continual training on curated data mixtures remain poorly understood, particularly for less represented languages such as Italian. To the best of our knowledge, OLMo et al. [3] is the only work specifically addressing the impact of data composition in an open-source setting; however, it is limited to the English language.

In this work, we address this gap by systematically investigating how incorporating high-quality data mixtures during continual pretraining affects model performance on English- and Italian-language tasks. A particular focus is placed on cultural knowledge evaluation, where curated data is expected to play a crucial role in enriching the model's ability to answer questions about Italian cultural content. To this end, we build on the Minerva-7B base model [10], a fully open-source LLM pretrained on a balanced corpus of Italian and English data (50% each), which provides a suitable foundation for evaluating bilingual continual pretraining strategies.

Specifically, we define three distinct high-quality data recipes for continual pretraining, varying in data dimen-

CLiC-it 2025: Eleventh Italian Conference on Computational Linguistics, September 24 — 26, 2025, Cagliari, Italy
*Corresponding author.
moroni@diag.uniroma1.it (L. Moroni); bonomo@diag.uniroma1.it (T. Bonomo); gioffre@diag.uniroma1.it (L. Gioffré); xu@diag.uniroma1.it (L. Xu); fedele@babelscape.com (D. Fedele); colosi@babelscape.com (L. Colosi); bejgu@babelscape.com (A. S. Bejgu); scire@babelscape.com (A. Scirè); navigli@diag.uniroma1.it (R. Navigli)

sions and source types, using both Italian and English texts. These include content rich in mathematical reasoning, encyclopedic knowledge, and copyrighted books. Through ablation studies, we examine the individual contribution of specific data sources—such as copyrighted material and mathematical content—on downstream performance across English and Italian benchmarks.

Additionally, we explore the effect of extending the model's maximum context length during continual pretraining, aiming to assess its impact on long-context understanding. After pretraining, we instruction-tune the various model variants using a bilingual (English and Italian) instruction-following dataset to evaluate their performance in conversational settings.

Finally, to properly evaluate the influence of longer context and data composition, we introduce INDAQA, a novel Italian benchmark for narrative question answering (Section 6.1). Using INDAQA, we demonstrate the benefits of longer context windows and specific high-quality data sources for complex language understanding tasks.

2. Related Work

Continual Training. Following the initial pretraining phase over trillions of tokens, it is now common practice to introduce high-quality data in a subsequent training stage to further enhance LLM performance and steer the model's distribution toward more controlled domains. Recent research has increasingly focused on continual pretraining as a practical and impactful approach. For instance, OLMo et al. [3] and Grattafiori et al. [4] introduce a mid-training stage that incorporates high-quality datasets into the pretraining process, e.g. GSM8K training set for mathematical reasoning. This stage is treated as a continuation of the initial training, employing an annealing learning rate that decays linearly to zero. This approach has been shown to improve downstream performance in tasks requiring structured reasoning and encyclopedic knowledge recall.

Continual training is also frequently employed to adapt released open-weight LLMs to specific languages or domains, thereby improving performance on targeted tasks. Basile et al. [11] and others demonstrate that adapting pretrained multilingual models to Italian using curated high-quality data leads to significant improvements in Italian-language benchmarks. Despite these advances, there is still a lack of systematic studies that ablate and isolate the specific contributions of different data mixing strategies in the continual pretraining stage—particularly for less represented languages like Italian. In our work, we assess the impact of controlled data used in the continual-pretraining stage, looking at their impact on English and Italian performance.

Context Length Manipulation. Large Language Models are typically pretrained with a fixed maximum context length, which limits the number of tokens they can process in a single sequence. Recent work by Xiong et al. [12] demonstrates how expanding the context length of Llama-2 models–from 4,096 to 32,728 tokens–can improve performance on long-context tasks. A critical aspect of long-context training is the choice of positional encoding. Most modern LLMs employ Rotary Positional Embeddings (RoPE) [13], which encode token positions by rotating the query and key vectors in attention layers. This approach maintains relative positional information and can be adapted for longer sequences. Recent studies show that modifying the RoPE base frequency during continual pretraining enables models to handle longer contexts and even extrapolate beyond the trained sequence lengths [14, 15]. Building on these findings, several recent LLMs have been released with extended context capabilities. For example, Grattafiori et al. [4] increases the context length of Llama-3 models from 8,192 to 128,000 tokens in the final stages of pretraining. Similarly, the Qwen model family [16] mostly supports contexts up to 32,000 tokens. However, despite these advancements, to the best of our knowledge, this paper is the first that systematically investigates the impact of context length manipulation on Italian-language tasks.

Evaluation of LLMs in Italian. Several recent efforts aim to close the evaluation gap between English and Italian for generative LLMs. One of the first initiatives, Ita-Bench [17], combines translated benchmarks with natively authored Italian tasks, focusing on instruction-following and question answering. Along the same lines, Magnini et al. [18] reframes native Italian resources into both multiple-choice and open-ended formats, studying the role of prompting strategies. More recently, ITALIC [19] introduces a multiple-choice question answering dataset entirely written in Italian, covering linguistic, cultural, and domain-specific knowledge. In parallel, Puccetti et al. [20] adapts Invalsi assessments to probe LLMs' multi-domain abilities.

Complementing these Italian-specific efforts, multilingual benchmarks have also emerged. Global-MMLU [21] extends MMLU to multiple languages via professional translation and cultural adaptation, while MultiLOKO [22] provides culturally grounded questions authored directly in each target language, including Italian. While these benchmarks cover a variety of linguistic and cultural aspects, they primarily focus on short-form tasks. Yet, many real-world scenarios, such as narrative comprehension and document-level reasoning, require models to process and integrate information across longer contexts. However, evaluation resources in Italian remain limited in this dimension. To fill this gap, we introduce INDAQA (Section 6.1), the first narrative question

answering benchmark designed to evaluate long-context comprehension in Italian.

3. Methodology

This work investigates the impact of continual training and the influence of different data sources on downstream performance, with particular attention to copyrighted material. Additionally, we aim to address a gap in the literature regarding the effect of context length expansion on performance in Italian.

We focus on three key dimensions:

- **Data recipes:** we introduce three distinct recipes designed to evaluate the role of data composition during continual training.
- **Context length:** we describe how we adapt models to long-context scenarios, using a selected data mixture from the previous step.
- **Instruction following:** we examine the instruction-following capabilities developed on top of each training recipe.

3.1. Data Recipes for Wide Linguistic Coverage

To evaluate the impact of various data sources on the continual training of an open-source LLM, namely Minerva-7B base model, we define several data recipes, each representing a distinct mixture of training corpora. Table 1 presents the data composition for one such configuration, which we refer to as **Recipe-1**[1]. This recipe incorporates a diverse set of sources. For Italian, we include: the Italian Wikipedia (Hugging Face version, 2023 dump, Italian split)[2] encyclopedic collection of text, RedPajama [23], a web-based collection, and Ita-Bench [17], a suite of Italian and English benchmarks for generative models (Italian training split). Regarding English, the dataset comprises: Wikipedia (English split), Ita-Bench (English training split), Fineweb-edu [24], a web-based collection, Project Gutenberg,[3] which comprises public-domain books, and FLAN [25, 26, 27, 28, 29], which contains different instructions for mathematical and logical reasoning.

Building on Recipe-1, we design two additional data mixtures, **Recipe-2** and **Recipe-3**, to evaluate the impact of mathematical reasoning data and the inclusion of a large volume of copyrighted books. Table 2 shows the data composition for these two recipes. Starting from the foundation of Recipe-1, we replace the standard Wikipedia dump with a curated and cleaned version collected by us, updated to May 2024. We also expanded the dataset with additional sources. For Italian, we included the Wikisource[4] collection of articles, Gazzetta Ufficiale,[5] which contains legislative and administrative acts of the Italian State, and Project Gutenberg. For English, we incorporated subsets of the Dolmino-mix dataset, used in the continual training of OLMo-2 [3], specifically the MATH and StackExchange (SE) components.

The key distinction between Recipe-2 and Recipe-3 is that Recipe-3 incorporates the Books3 dataset [30], which allows the impact of including closed-copyrighted book content to be quantified. Further details on our data preprocessing steps can be found in Appendix B.

Data Source	Tokens	Times	Final Tokens
Italian			
Benchmarks	6.9M	21	144M
Wikipedia	814M	3	2.4B
RedPajama	5.8B	2	11.6B
English			
Benchmarks	55M	5	275M
Wikipedia	2.4B	3	7.3B
Fineweb-edu	6B	2	12B
Gutenberg	1B	1	1B
FLAN	9.5B	1	9.5B
Code			
The Stack	3.3B	1	3.3B
Recipe-1	-	-	47.9B

Table 1
Breakdown of the data components of Recipe-1. *Times* refer to the number of times each data source is sampled.

3.2. Long-context Adaptation

Recent studies demonstrate that continual pre-training can substantially extend the context length of LLMs [12, 31]. Based on previous work and motivated by the lack of a proper assessment of context expansion in Italian, we carry out the context length expansion on Recipe-3, our continually pre-trained model described in Section 3.1. Following the methodology of Xiong et al. [12], we extend the maximum context length from 4,096 tokens (the original limit of Minerva-7B) to 16,384 tokens. This expansion requires adjusting the Rotary Position Embedding (RoPE) base frequency θ from 10,000 to 500,000 to accommodate the increased sequence length. To establish baseline comparisons, we adjust the RoPE base frequency in our continually-trained models obtained through the recipes of Section 3.1 in order to adapt them to longer contexts.

[1] Recipe-1 corresponds to the continual pretraining data used in the first version of the released Minerva-7B.
[2] https://huggingface.co/datasets/wikimedia/wikipedia
[3] https://huggingface.co/datasets/manu/project_gutenberg
[4] https://huggingface.co/datasets/wikimedia/wikisource
[5] https://huggingface.co/datasets/mii-llm/gazzetta-ufficiale

Data Source	Tokens	Times	Final Tokens
Italian			
Benchmarks	6.9M	7	50M
Wikisource	53M	5	266M
RedPajama	20B	2	40B
Gazzetta	853M	1	853M
Gutenberg	100M	5	500M
Wikipedia	1.2B	5	6.1B
English			
Benchmarks	55M	5	275M
Fineweb-edu	4.3B	2	8.6B
FLAN	12B	1	12B
Wikipedia	7.1B	1	7.1B
$\text{Dolmino}_{\text{MATH}}$	11.7B	1	11.7B
$\text{Dolmino}_{\text{SE}}$	1.5B	1	1.5B
Books3	24B	1	24B
Code			
The Stack	2.5B	1	2.5B
Recipe-2	-	-	92B
Recipe-3	-	-	116B

Table 2
Breakdown of the data components of Recipe-2 and Recipe-3. Recipe-3 builds on Recipe-2, adding Books3. *Times* refer to the number of times each data source is sampled.

3.3. Instruction Following

After continual pre-training, each recipe is converted into an *instruct* model through an SFT stage on the dialogue mixture summarised in Table 3. We base the mixture on TÜLU-v3 [9], a popular open-source 940K-conversation corpus covering 85 task families (reasoning, code, function-calling, safety, tool use, etc.) mined from public APIs and manually filtered for policy compliance, which provides the broad, structured competence expected of modern assistants. To inject high-signal, stylistically polished examples we add the 1000-turn LIMA dataset [8] and its Italian counterpart LIMA-IT, produced by us by translating every prompt/response pair with GPT-4o-mini under a fidelity-preserving prompt; this gives the model a high-quality set of concise, helpful dialogue in both languages. We expand our selection with additional Italian-centric datasets: i) WildChat-IT, consisting of 5K informal prompts; ii) TowerBlocks-v0.2, containing 7K bilingual it-en public-service Q&A pairs; iii) GPT-4o-ITA-Instruct, with 15K high-quality synthetic chain-of-thought examples; and iv) Aya, which includes 700 role-play and reasoning turns, specifically targeting colloquial language, public administration knowledge, and culturally grounded reasoning.

Dataset	Language(s)	# Instructions
TÜLU-v3	EN	940 000
LIMA	IT/EN	2 000
WildChat-IT	IT	5 000
TowerBlocks-v0.2	IT/EN	7 276
GPT-4o-ITA-Instruct	IT	15 000
Aya	IT	700

Table 3
Overview of the SFT datasets used for instruction tuning.

4. Experimental setup

4.1. Continual training

We trained the Minerva-7B base model using three different data recipes, as detailed in Section 3.1. For each recipe, we performed continual pretraining using a newly initialized optimizer–namely AdamW [32]. Across all recipes, we used a batch size of 1024 and a maximum context length of 4096 tokens, consistent with the original pretraining setup. The learning rate was set to a maximum of 1×10^{-5}, with a warmup period of 200 steps for Recipe-1 and 600 steps for Recipe-2 and Recipe-3, reflecting the larger token volumes in the latter two.

For the extended context training variant of Recipe-3, which we name $\textbf{Recipe-3}_{\textbf{16K}}$, we aimed to maintain consistent training dynamics by keeping the number of gradient updates fixed across both the standard and long-context regimes. Specifically, when increasing the context length from 4,096 to 16,384 tokens (a 4× increase), we proportionally reduced the batch size by a factor of 4. This ensured that each gradient update processed approximately the same total number of tokens, allowing for a controlled comparison between standard continual training and long-context adaptation.

We ran our continual training experiments through the `LLM-Foundry`[6] library. Each run used 64 custom NVIDIA-A100 with 64GB of VRAM, scattered on 16 nodes. All the experiments were executed on the Leonardo supercomputer[7].

4.2. Instruction finetuning

Supervised fine-tuning was carried out with the `LLAMA-Factory`[8] toolkit, which supports several conversation templates and provides utilities for efficient data parallelization. We fine-tuned the *full* Minerva-7B weights (no LoRA/adapters) in bfloat 16 mixed precision. Training lasted two epochs with a peak learning rate of 1×10^{-6} scheduled by cosine decay after a 10% warm-up, and AdamW as the optimizer. We used an

[6] https://github.com/mosaicml/llm-foundry
[7] https://www.hpc.cineca.it/systems/hardware/leonardo/
[8] https://github.com/hiyouga/Llama-Factory

effective batch of 64 sequences ($\approx 128k$ tokens). All models were trained with a 4096-token context window, except the long-context variant of Recipe-3, which retained its 16384-token window. End-to-end, each recipe consumed about 210 GPU-hours (240 for the long-context run). Detailed timing and CO_2 estimates are shown in Appendix A.

5. Evaluation

5.1. Language Modeling by Genre

To evaluate the impact of the different data recipes, we analyze perplexity scores of trained LLMs on held-out data from various genres. Specifically, we test the models on three distinct genres: Books, Wikipedia, and News.

The Books set consists of 51 held-out books selected from Books3 [30], covering 25 different genres, in English languages. The Wikipedia set includes 50 Italian pages from a 2025 snapshot[9], excluded from the training data used in all recipes. The News set consists of 200 Italian newspaper articles we independently collected from 2025 publications, ensuring they were never seen during any training step. Table 4 reports the language modeling performance, measured by perplexity, across these domains for each trained model.

Regarding Books, incorporating Books3 into the training mix significantly lowers perplexity, as seen in the improved performance of Recipe-3. This indicates that including in-domain book content enhances generalization to literary-style text. Additionally, testing Recipe-3_{16K} using 16k context on Books drops the perplexity to 8.98, further improving modeling on extended sequences.

For the Wikipedia genre, all three recipes outperform the original pretrained model, demonstrating improved ability to model high-quality encyclopedic text. Notably, Recipe-2 and Recipe-3 achieve the lowest perplexity, suggesting benefits from training on more recent and cleaner Wikipedia texts.

In contrast, for the News genre, perplexity differences among the recipes are minimal (±0.20), indicating a limited impact of the training data variations on this domain. Interestingly, the base model achieves the lowest perplexity.

Bottom line: *The modeling of literary-style texts and Wikipedia articles is influenced by the choice of continual pretraining strategies, whereas News articles show no differences.*

[9]We process the May 1st, 2025 Wikipedia dump by first discarding pages with fewer than 500 tokens, and then sampling uniformly at random from the resulting set.

Model	Books ↓	Wikipedia ↓	News ↓
Pretraining	$11.05_{\pm 0.55}$	$7.54_{\pm 0.36}$	$\mathbf{10.05_{\pm 0.22}}$
Recipe-1	$11.08_{\pm 0.55}$	$7.20_{\pm 0.37}$	$10.22_{\pm 0.22}$
Recipe-2	$12.12_{\pm 0.62}$	$6.78_{\pm 0.41}$	$10.45_{\pm 0.23}$
Recipe-3	$9.57_{\pm 0.48}$	$\mathbf{6.72_{\pm 0.41}}$	$10.45_{\pm 0.23}$
Recipe-3_{16K}	$\mathbf{9.56_{\pm 0.48}}$	$6.75_{\pm 0.41}$	$10.42_{\pm 0.23}$

Table 4
Perplexity scores of our proposed training recipes on heldout, comprising texts from the following genres: Books, Wikipedia and News. The input text is truncated to 4K tokens.

5.2. Multi-Choice Question Answering

To properly assess how different continual pretraining recipes influence LLM capabilities, we evaluate our trained models on a range of Italian-language benchmarks. In this Section, we focus exclusively on the continually-trained models, before applying any instruction tuning. This approach isolates the effects of continual pretraining and avoids biases introduced by SFT data. We conduct evaluations using the `LM-Evaluation-Harness` [33] library, leveraging the multi-choice format: a model's next-token prediction is used to assess its QA ability.

We evaluate the models using ITA-Bench [17], selecting a diverse set of tasks from the benchmark: AMI (Misogyny Detection), GhigliottinAI (GH; a culturally grounded game), NERMUD (Named Entity Recognition), Prelearn (PL; Prerequisite Learning), ARC (Scientific Reasoning), BoolQ (BQ; Boolean Questions), GSM8K (Mathematics), HellaSwag (HS; Textual Entailment), MMLU (Multi-domain QA), PIQA (Physical Interaction QA), and SCIQ (Science Questions). For AMI, GhigliottinAI, and NERMUD, we use ITA-Bench's cloze-style evaluation format.

Table 5 shows that all recipes of continual pretraining consistently improve over the pretrained model, with an average gain of approximately +5.0 points. This result reinforces the importance of continual pretraining on high-quality (e.g., Wikipedia, Fineweb-edu) and synthetic datasets (e.g., FLAN, Dolmino-MATH subset). Notably, MMLU exhibits substantial improvements across all recipes ($\approx +15$ points), highlighting strong generalization on multi-domain QA tasks. The best average performance is achieved by Recipe-2 and the long-context variant of Recipe-3. Recipe-1 underperforms, particularly on math-related benchmarks such as ARC and GSM8K, indicating the critical role of domain-specific data (e.g., Dolmino-MATH) in boosting model capabilities.

Bottom line: *Continual pretraining consistently boosts downstream performance; mathematical data improves STEM QA, while copyrighted books have minimal impact.*

Recipe	AMI 0-shot	GH 5-shot	NERMUD 0-shot	PL 5-shot	ARC_C 5-shot	BQ 0-shot	GSM8K 0-shot	HS 0-shot	MMLU 5-shot	PIQA 0-shot	SCIQ 0-shot	AVG -
Pretraining	45.23	45.75	59.99	54.88	39.49	59.65	52.31	60.41	25.45	**70.2**	90.36	54.88
Recipe-1	49.55	44.85	41.77	59.38	42.49	**82.66**	51.25	**62.50**	40.79	68.48	90.76	57.68
Recipe-2	**54.56**	**46.84**	51.24	54.87	**43.37**	80.76	54.28	60.70	41.23	68.42	90.25	**59.85**
Recipe-3	52.65	40.87	45.26	**61.75**	**43.37**	80.76	54.36	61.42	41.56	68.42	90.86	58.24
Recipe-3_{16K}	51.43	40.14	**62.29**	57.12	41.52	82.20	**54.81**	60.96	**41.63**	68.18	**92.28**	59.57

Table 5
Evaluation of our proposed continual training recipes on ITA-Bench. Specifically, we report 0- and 5-shot accuracy scores on each task on ITA-Bench.

5.3. Mathematical Evaluation

To assess the impact of different continual-pretraining recipes on math capabilities, we rely on two widely used English mathematical benchmarks: GSM8k [34] and MATH [35]. The former contains grade school math word problems, while the latter comprises challenging competition mathematics problems. We evaluate our models using the `LM-Evaluation-Harness` [33], using its implementations of both benchmarks. For GSM8k, we adopt an 8-shot Chain-of-Thought prompting setup, while for MATH, we follow the Minerva-MATH [36] protocol, using 4-shot Chain-of-Thought prompting. Both benchmarks use the `generate_until` setup, with model outputs evaluated via post-processing for accuracy. We compare our recipes to different open-source Italian (occiglot-7b-it-en-instruct[10], ANITA-8B [37]) and multilingual (Llama-3.1-8B [4], Mistral-7B [38], Qwen3-8B [39]) models, all in the same parameter range.

Table 6 presents the results of tested models, with our four continually pre-trained Minerva models evaluated both before and after instruction tuning. On GSM8k, Recipe-2 achieves the highest accuracy in both settings, followed by Recipe-3, while Recipe-1 consistently underperforms. Instruction tuning yields consistent improvements across all recipes, reinforcing the overall ranking and demonstrating its positive effect. These findings suggest that incorporating mathematical data, such as Dolmino-MATH, during continual pre-training plays a significant role in enhancing mathematical reasoning. For the MATH dataset, Recipes 2 and 3 outperform Recipe-1 in the base (pre-instruction tuning) setting, particularly benefiting from long-context capabilities. Interestingly, after instruction tuning, the performance gap narrows, with Recipe-1 becoming more competitive.

When comparing Minerva models to state-of-the-art systems on GSM8k, they lag behind closed-data models in both Italian and English. On the MATH dataset, Minerva is comparable to Occiglot and Mistral, two closed-data models, but still lags behind top-performing English-centric systems. This highlights the perfomance gap that Italian open-data LLMs must bridge.

[10]https://huggingface.co/occiglot/occiglot-7b-it-en-instruct

Model	MATH	GSM8k
Minerva Base Models		
Recipe-1	2.48	14.70
Recipe-2	9.57	**34.42**
Recipe-3	8.96	26.45
Recipe-3_{16K}	**10.26**	32.29
Minerva Instruct Models		
Recipe-1	10.14	24.63
Recipe-2	12.84	**42.45**
Recipe-3	**13.00**	37.98
Recipe-3_{16K}	12.82	40.25
Italian-specific Models		
Occiglot-7b	10.86	49.88
ANITA-8B	17.56	60.65
English-first Models		
Llama-3.1-8B	41.94	80.66
Mistral-7B-v0.3	13.92	53.22
Qwen3-8B	65.00	87.86

Table 6
Mathematical evaluation results on different Minerva continual pre-training recipes (before and after instruction fine-tuning) and State-of-the-Art models on Minerva-MATH (4-shot) with sub-categories, and GSM8k (8-shot).

Bottom line: *Continual pretraining on mathematical data consistently improves accuracy on math problems. Instruction tuning on TULU-v3 helps mitigate the shortcomings of Recipe-1 on the MATH benchmark.*

5.4. Cultural Evaluation

We assess the impact of our recipes used during continual pre-training by leveraging the Italian part of the Multiloko [22] dataset (250 instances), which provides questions on cultural content along with multiple acceptable answers. We then compare our continually pre-trained and instruction finetuned Minerva models to other Italian and English models, as in the previous section.

According to the results in Table 7, Recipe-1 is the best performing model, both in Zero- and Few-Shot settings, surpassing both the Italian-specific and the English-centric counterparts.

Model	MultiLoKo				ITALIC-GEN	
	0-shot		5-shot		0-shot	5-shot
	EM	F1	EM	F1	METEOR	
Minerva Models						
Recipe-1	**0.17**	**0.27**	**0.18**	**0.29**	**0.24**	**0.27**
Recipe-2	0.07	0.16	0.12	0.22	0.20	0.23
Recipe-3	0.13	0.23	0.13	0.23	0.21	0.22
Recipe-3_{16K}	0.11	0.20	0.13	0.24	0.22	0.23
Italian-specific Models						
occiglot-7b	0.14	0.21	0.10	0.15	0.22	0.20
ANITA-8B	0.14	0.18	0.13	0.17	0.21	0.15
English-first Models						
Llama-3.1-8B	0.15	0.20	0.11	0.15	0.21	0.20
Mistral-7B-v0.3	0.06	0.14	0.08	0.16	0.15	0.19
Qwen3-8B	0.09	0.14	0.08	0.13	0.16	0.19

Table 7
Cultural alignment results on Multiloko Italian and ITALIC-GEN datasets. We report 0- and 5-shot EM and F1 Scores for Multiloko, while METEOR metric is used for ITALIC-GEN.

Recipe-2 and Recipe-3, which are trained on a large amount of mathematics, code, and English-copyrighted books, do not show the same cultural alignment in the MultiLoKo Italian set. This observation demonstrates that synthetic, mathematical, and English literary data can be detrimental for Italian cultural alignment.

Recently, Seveso et al. [19] have shown that Italian-first models perform consistently lower than English-first ones on the ITALIC dataset. We hypothesize that the multiple choice format could be particularly problematic and might obscure the cultural knowledge recall of language models. Therefore, we examine whether these results hold when reframing ITALIC in an open-ended setting, which better reflects potential use cases for generative models. Details on how we reframed the dataset, ITALIC-GEN, are in Appendix D.

We use METEOR [40] to evaluate the performance, as only one reference answer per question is available, and standard string matching metrics, such as EM, may struggle when model outputs and references differ significantly in phrasing and/or length. The results in Table 7 confirm the trend seen in MultiLoKo, which again demonstrates the cultural alignment capacity of Minerva models. Our results further suggest that incorporating structured mathematical data during pretraining can constrain a model's acquisition of cultural knowledge.

Bottom line: *Multiple-choice QA may not be well suited for evaluating cultural competence, as it limits expressive freedom and fails to capture the nuanced reasoning required for culturally-grounded responses. Notably, Italian-native models emerge to be the most aligned with Italian culture, highlighting the importance of language-specific pretraining.*

6. Long-context Evaluation on Narrative Text

To evaluate the long-context capabilities of our model, we focus on narrative question answering, a task that requires the processing and understanding of extensive narrative text in order to answer questions. NarrativeQA [41], a widespread benchmark for this task, was constructed in English, which limits its use for the evaluation of long-context performance in other languages. To address this limitation, we introduce INDAQA (Section 6.1), a novel benchmark for Italian narrative question answering, and, to the best of our knowledge, the first narrative question answering dataset in Italian. We describe the evaluation setup for base and instruction-tuned models on both NarrativeQA and INDAQA in Section 6.2 and report the results in Section 6.3.

6.1. INDAQA - Italian Narrative DAtaset for Question Answering

We start building the dataset from the Italian split of Echoes from Alexandria [42], collecting 365 (book, summary) pairs with full texts from Wikisource and summaries from Wikipedia. After manually verifying alignment and removing plot-unrelated content from summaries, we prompt an LLM[11] to generate 20 question-answer pairs per book using the following guidelines: (i) questions must be unique, (ii) questions must be clear, unambiguous, and answerable from the summary alone, and (iii) each question requires having two short, potentially different, reference answers.

After gathering a large number of samples, we filter them through three sequential steps. First, we deduplicate questions, but rather than discarding duplicates entirely, we retain all unique answers as additional references for the remaining samples. We also preserve different reformulations of identical questions, as NarrativeQA contains similar variations. Second, we remove *unanswerable questions*, i.e., samples containing invalid responses such as *"Information not present in the summary."* Finally, we filter out *meta-questions* that focus on structural rather than plot elements (e.g., *"What happens in chapter 3?"* or *"What is the title of the book?"*). The last two filtering steps are carried out through a set of manually derived RegEx patterns. Examples of samples that were filtered out are showcased in Table 11 (Appendix).

We reduce the average answer length so as to be better aligned with NarrativeQA by employing an LLM to shorten the replies. We perform this step only for the samples having no reference answers with less than 5 tokens. The final statistics on the QA length are presented in Table 8. We manually validate generation and

[11]We use Gemini-2.0-Flash and Gemini-2.0-Flash-Lite.

Metric	Avg. Length (Tokens)	# Samples
	NarrativeQA	
Question	8.60 ± 3.30	10,557
1st Answer	4.55 ± 3.91	10,557
2nd Answer	3.89 ± 3.30	10,557
	INDAQA	
Question	7.06 ± 2.14	13,757
1st Answer	2.88 ± 1.27	13,757
2nd Answer	5.16 ± 2.70	13,669
3rd Answer	9.27 ± 3.41	4,180
4th Answer	7.40 ± 2.26	514
5th Answer	9.61 ± 2.66	251

Table 8
Statistics on the length of the QA samples. The average length of the first and second answers are respectively less and on par with NarrativeQA average on the test set. Due to the described deduplication steps, some QA samples have up to 5 reference answers, while a small portion (88) have only 1 reference answer.

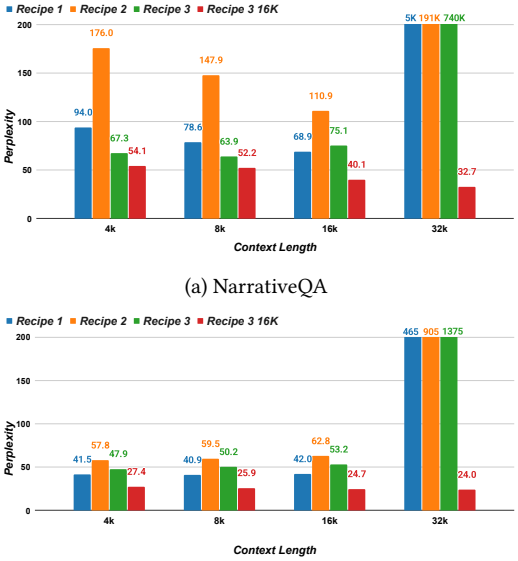
filtering steps on 17 documents (646 QA samples, 5% of the dataset) spanning diverse summary lengths (18-1200 tokens). Each sample is annotated for acceptability using the same criteria used for generation, yielding a 2.32% error rate after filtering.

Our final dataset, **INDAQA**, consists of texts with an average length shorter than NarrativeQA (27k vs 47k tokens) due to the prevalence of short stories and theatrical plays.[12] The size of the two datasets is comparable (365 vs 355 documents) with slightly more average QA samples in INDAQA (37.83 vs 29.74). We also report the type of questions in the dataset by analyzing the first few tokens of the questions in Table 10 (Appendix). More details can be found in Appendix C.

6.2. Long-context Evaluation Setup

Base-model evaluation To evaluate the effectiveness of our long-context continual training approach, we compare Recipe-3_{16K} against Recipe-1, Recipe-2 and Recipe-3. Except for Recipe-3_{16K}, we adapt each model to process longer sequences by tuning the RoPE base frequency to $\theta = 100{,}000$. We assess each model's ability to utilize extended local context using an adapted version of NarrativeQA and INDAQA. Specifically, we truncate each text at varying target context lengths (4,096, 8,192, 16,384 and 32,768 tokens), and we record the minimum perplexity achieved by each model across the ground-truth answers when given the truncated text and respective questions. We assume that models effectively processing long contexts will show lower perplexity on correct answers than those struggling with extended documents.

[12] In our experiments, the input text is always truncated at 16k tokens.



(a) NarrativeQA

(b) INDAQA

Figure 1: Evaluation of our long-context model (Recipe-3_{16K}) against the other recipes on (a) NarrativeQA and (b) INDAQA in terms of the average perplexity of correct answers to a question at varying context lengths.

Instruction-tuning evaluation We evaluate the instruction-tuned versions of the Minerva continual pretrained models alongside various systems, as in previous sections. Benchmarking is conducted on both NarrativeQA and INDAQA to assess real-world performance in English and Italian narrative question answering. We report METEOR [40] scores to measure answer quality against the reference responses. We truncate the book texts to 16,384 and 32,768 tokens to match our target context lengths, following the approach used in LongBench [43]. While some questions may require context that is excluded by this truncation, all models are affected equally, ensuring a fair comparison between them.

6.3. Results

In Figure 1 we present the results of our base-model evaluation. Our long-context adaptation of Recipe-3 clearly enables the model to achieve a lower perplexity on the answers of NarrativeQA and INDAQA at all context lengths tested, indicating an effective adaptation to long data. It is especially interesting to note the results at 32,768 tokens: adapting models continually trained with shorter context lengths through RoPE frequency tuning is not enough to avoid huge spikes in perplexity, while Recipe-3_{16K} is able to effectively model text at double its continual training context window.

	Model	Ctx len	M@16K	M@32K
NarrativeQA	*Minerva models*			
	Recipe-1	4K	13.7	3.2
	Recipe-2	4K	10.1	2.2
	Recipe-3	4K	12.9	2.4
	Recipe-3_{16K}	16K	<u>21.4</u>	<u>20.5</u>
	Italian-specific Models			
	occiglot-7b	32K	16.4	15.9
	ANITA-8B	8K	3.2	3.1
	English-first Models			
	Llama-3.1-8B	128K	**24.0**	**28.7**
	Mistral-7B-v0.3	32K	21.7	25.6
INDAQA	*Minerva models*			
	Recipe-1	4K	17.3	11.1
	Recipe-2	4K	12.2	7.3
	Recipe-3	4K	13.5	8.3
	Recipe-3_{16K}	16K	**25.9**	<u>26.0</u>
	Italian-specific Models			
	occiglot-7b	32K	19.9	19.9
	ANITA-8B	8K	7.5	7.0
	English-first Models			
	Llama-3.1-8B	128K	<u>24.9</u>	**29.3**
	Mistral-7B-v0.3	32K	22.5	27.7

Table 9
Continual pre-training recipe evaluation on NarrativeQA and INDAQA after instruction fine-tuning. M@16k and M@32k denote METEOR scores with 16,384 and 32,768 token book contexts. Bold scores indicate best overall performance; underlined scores indicate best Italian-specific model.

Table 9 presents the results of the evaluation of our instruction-tuned models. As expected, Recipe-3_{16K} achieves higher results on all settings, surpassing Recipe-1 on all experiments with books truncated to 16k tokens by 7.7 points on NarrativeQA and 8.6 on INDAQA. The difference is even larger when we extend the truncation of books to 32K tokens, with Recipe-3_{16K} achieving 17.3 and 14.9 more METEOR points in NarrativeQA and INDAQA, respectively.

Minerva models perform comparably to other models of the same size, both Italian-specific (occiglot-7b-it-en-instruct[13], ANITA-8B [37]) and multilingual (Llama-3.1-8B [4], Mistral-7B [38]). On NarrativeQA, the Recipe-3_{16K} variant achieves a METEOR score of 21.4 and 20.5 at a context length of 16K and 32K respectively, ranking behind Llama-3.1 and Mistral-v0.3. In contrast, the Minerva model continually pre-trained with Recipe-3_{16K} outperforms all tested models on INDAQA at 16K tokens of context, achieving the highest METEOR score of 25.9. At 32K tokens of context, it ranks second only to Llama-3.1 and Mistral-v0.3, scoring 3.3 and 1.7 points lower respectively on the METEOR metric. This performance gap is expected, given that Recipe-3_{16K}'s continual training was conducted at half the context length (16K tokens).

Bottom line: *Extending context length to 16K tokens via continual pre-training improves modeling capabilities over training-free methods and enhances robustness at 32K tokens. Recipe-3_{16K} achieves strong narrative QA performance in both English and Italian, outperforming Italian-specific models and matching English-first LLMs.*

7. Conclusion

This work explores the impact of data mixing strategies and long-context expansion on Italian language modeling. We conduct continual pretraining using three distinct data recipes and apply a unified instruction-following fine-tuning approach to all resulting models. Our evaluation assesses language modeling capabilities on genre-specific data, highlighting that copyrighted books included in the training recipes reduce perplexity on literary texts. We benchmark the proposed continual pretraining recipes across several multi-domain tasks, with a focus on mathematical reasoning, demonstrating that genre-specific data, such as mathematical texts and high-quality web content contribute to overall performance improvements, whereas copyrighted books do not consistently offer the same benefit. We also investigate cultural alignment, finding that English datasets, such as mathematical texts and English-copyrighted books, can negatively impact performance on culturally-aware Italian-specific tasks. Additionally, our ITALIC-GEN adaptation offers a complementary perspective on cultural evaluation, uncovering encouraging results for Italian LLMs. Lastly, we evaluate long-context capabilities through narrative question answering in both English and Italian. Due to the absence of an Italian benchmark, we introduced INDAQA, a new dataset for Italian narrative QA, and show that extending the context length of a model consistently improves its downstream performance on narrative QA.

Acknowledgments

Luca Moroni, Tommaso Bonomo, Luca Gioffré and Lu Xu gratefully acknowledge the support of the AI Factory IT4LIA project. Roberto Navigli acknowledges the support of the PNRR MUR project PE0000013-FAIR. We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy).

[13] https://huggingface.co/occiglot/occiglot-7b-it-en-instruct

References

[1] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, O. Vinyals, J. W. Rae, L. Sifre, Training compute-optimal large language models, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22, Curran Associates Inc., Red Hook, NY, USA, 2022.

[2] D. Groeneveld, I. Beltagy, E. Walsh, A. Bhagia, R. Kinney, O. Tafjord, A. Jha, H. Ivison, I. Magnusson, Y. Wang, S. Arora, D. Atkinson, R. Authur, K. Chandu, A. Cohan, J. Dumas, Y. Elazar, Y. Gu, J. Hessel, T. Khot, W. Merrill, J. Morrison, N. Muennighoff, A. Naik, C. Nam, M. Peters, V. Pyatkin, A. Ravichander, D. Schwenk, S. Shah, W. Smith, E. Strubell, N. Subramani, M. Wortsman, P. Dasigi, N. Lambert, K. Richardson, L. Zettlemoyer, J. Dodge, K. Lo, L. Soldaini, N. Smith, H. Hajishirzi, OLMo: Accelerating the science of language models, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 15789–15809. URL: https://aclanthology.org/2024.acl-long.841/. doi:10.18653/v1/2024.acl-long.841.

[3] T. OLMo, P. Walsh, L. Soldaini, D. Groeneveld, K. Lo, S. Arora, A. Bhagia, Y. Gu, S. Huang, M. Jordan, N. Lambert, D. Schwenk, O. Tafjord, T. Anderson, D. Atkinson, F. Brahman, C. Clark, P. Dasigi, N. Dziri, M. Guerquin, H. Ivison, P. W. Koh, J. Liu, S. Malik, W. Merrill, L. J. V. Miranda, J. Morrison, T. Murray, C. Nam, V. Pyatkin, A. Rangapur, M. Schmitz, S. Skjonsberg, D. Wadden, C. Wilhelm, M. Wilson, L. Zettlemoyer, A. Farhadi, N. A. Smith, H. Hajishirzi, 2 olmo 2 furious, 2025. URL: https://arxiv.org/abs/2501.00656. arXiv:2501.00656.

[4] A. 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, The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[5] Y. Xie, K. Aggarwal, A. Ahmad, Efficient continual pre-training for building domain specific large language models, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 10184–10201. URL: https://aclanthology.org/2024.findings-acl.606/. doi:10.18653/v1/2024.findings-acl.606.

[6] L. Moroni, G. Puccetti, P.-L. Huguet Cabot, A. S. Bejgu, A. Miaschi, E. Barba, F. Dell'Orletta, A. Esuli, R. Navigli, Optimizing LLMs for Italian: Reducing token fertility and enhancing efficiency through vocabulary adaptation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Findings of the Association for Computational Linguistics: NAACL 2025, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 6646–6660. URL: https://aclanthology.org/2025.findings-naacl.371/.

[7] N. Ding, Y. Chen, B. Xu, Y. Qin, S. Hu, Z. Liu, M. Sun, B. Zhou, Enhancing chat language models by scaling high-quality instructional conversations, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 3029–3051. URL: https://aclanthology.org/2023.emnlp-main.183/. doi:10.18653/v1/2023.emnlp-main.183.

[8] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, S. Zhang, G. Ghosh, M. Lewis, L. Zettlemoyer, O. Levy, Lima: Less is more for alignment, 2023. URL: https://arxiv.org/abs/2305.11206. arXiv:2305.11206.

[9] N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, S. Lyu, Y. Gu, S. Malik, V. Graf, J. D. Hwang, J. Yang, R. L. Bras, O. Tafjord, C. Wilhelm, L. Soldaini, N. A. Smith, Y. Wang, P. Dasigi, H. Hajishirzi, Tulu 3: Pushing frontiers in open language model post-training, 2025. URL: https://arxiv.org/abs/2411.15124. arXiv:2411.15124.

[10] R. Orlando, L. Moroni, P.-L. Huguet Cabot, S. Conia, E. Barba, S. Orlandini, G. Fiameni, R. Navigli, Minerva LLMs: The first family of large language models trained from scratch on Italian data, in: F. Dell'Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 707–719. URL: https://aclanthology.org/2024.clicit-1.77/.

[11] P. Basile, E. Musacchio, M. Polignano, L. Siciliani, G. Fiameni, G. Semeraro, Llamantino: Llama 2 models for effective text generation in italian language, 2023. URL: https://arxiv.org/abs/2312.09993. arXiv:2312.09993.

[12] W. Xiong, J. Liu, I. Molybog, H. Zhang, P. Bhargava, R. Hou, L. Martin, R. Rungta, K. A. Sankararaman, B. Oguz, M. Khabsa, H. Fang, Y. Mehdad, S. Narang, K. Malik, A. Fan, S. Bhosale, S. Edunov, M. Lewis, S. Wang, H. Ma, Effective long-context

scaling of foundation models, in: K. Duh, H. Gomez, S. Bethard (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 4643–4663. URL: https://aclanthology.org/2024.naacl-long.260/. doi:10.18653/v1/2024.naacl-long.260.

[13] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, Y. Liu, Roformer: Enhanced transformer with rotary position embedding, Neurocomputing 568 (2024) 127063. URL: https://www.sciencedirect.com/science/article/pii/S0925231223011864. doi:https://doi.org/10.1016/j.neucom.2023.127063.

[14] X. Liu, H. Yan, C. An, X. Qiu, D. Lin, Scaling laws of rope-based extrapolation, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net, 2024. URL: https://openreview.net/forum?id=JO7k0SJ5V6.

[15] Y. Wu, Y. Gu, X. Feng, W. Zhong, D. Xu, Q. Yang, H. Liu, B. Qin, Extending context window of large language models from a distributional perspective, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 7288–7301. URL: https://aclanthology.org/2024.emnlp-main.414/. doi:10.18653/v1/2024.emnlp-main.414.

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

[17] L. Moroni, S. Conia, F. Martelli, R. Navigli, Towards a more comprehensive evaluation for Italian LLMs, in: F. Dell'Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 584–599. URL: https://aclanthology.org/2024.clicit-1.67/.

[18] B. Magnini, R. Zanoli, M. Resta, M. Cimmino, P. Albano, M. Madeddu, V. Patti, Evalita-llm: Benchmarking large language models on italian, 2025. URL: https://arxiv.org/abs/2502.02289. arXiv:2502.02289.

[19] A. Seveso, D. Potertì, E. Federici, M. Mezzanzanica, F. Mercorio, ITALIC: An Italian culture-aware natural language benchmark, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 1469–1478. URL: https://aclanthology.org/2025.naacl-long.68/.

[20] G. Puccetti, M. Cassese, A. Esuli, The invalsi benchmarks: measuring the linguistic and mathematical understanding of large language models in Italian, in: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert (Eds.), Proceedings of the 31st International Conference on Computational Linguistics, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 6782–6797. URL: https://aclanthology.org/2025.coling-main.453/.

[21] S. Singh, A. Romanou, C. Fourrier, D. I. Adelani, J. G. Ngui, D. Vila-Suero, P. Limkonchotiwat, K. Marchisio, W. Q. Leong, Y. Susanto, R. Ng, S. Longpre, W.-Y. Ko, S. Ruder, M. Smith, A. Bosselut, A. Oh, A. F. T. Martins, L. Choshen, D. Ippolito, E. Ferrante, M. Fadaee, B. Ermis, S. Hooker, Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation, 2025. URL: https://arxiv.org/abs/2412.03304. arXiv:2412.03304.

[22] D. Hupkes, N. Bogoychev, Multiloko: a multilingual local knowledge benchmark for llms spanning 31 languages, 2025. URL: https://arxiv.org/abs/2504.10356. arXiv:2504.10356.

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

[24] A. Lozhkov, L. Ben Allal, L. von Werra, T. Wolf, Fineweb-edu: the finest collection of educational content, 2024. URL: https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu. doi:10.57967/hf/2497.

[25] B. Goodson, Fine flan: Seqio to parquet so you don't have to, https://https://huggingface.co/datasets/Open-Orca/FLAN, 2023.

[26] S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei, A. Roberts, The flan collection: Designing data and methods for effective instruction tuning, 2023. arXiv:2301.13688.

[27] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, Q. V. Le, Finetuned language models are zero-shot learners, 2022. arXiv:2109.01652.

[28] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. Nayak, D. Datta, J. Chang, M. T.-J. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A. Fries, R. Teehan, T. Bers, S. Biderman, L. Gao, T. Wolf, A. M. Rush, Multitask prompted training enables zero-shot task generalization, 2022. arXiv:2110.08207.

[29] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Arunkumar, A. Ashok, A. S. Dhanasekaran, A. Naik, D. Stap, E. Pathak, G. Karamanolakis, H. G. Lai, I. Purohit, I. Mondal, J. Anderson, K. Kuznia, K. Doshi, M. Patel, K. K. Pal, M. Moradshahi, M. Parmar, M. Purohit, N. Varshney, P. R. Kaza, P. Verma, R. S. Puri, R. Karia, S. K. Sampat, S. Doshi, S. Mishra, S. Reddy, S. Patro, T. Dixit, X. Shen, C. Baral, Y. Choi, N. A. Smith, H. Hajishirzi, D. Khashabi, Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks, 2022. arXiv:2204.07705.

[30] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, C. Leahy, The pile: An 800gb dataset of diverse text for language modeling, 2020. URL: https://arxiv.org/abs/2101.00027. arXiv:2101.00027.

[31] Q. Team, Qwen2.5-1m: Deploy your own qwen with context length up to 1m tokens, 2025. URL: https://qwenlm.github.io/blog/qwen2.5-1m/.

[32] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. URL: https://arxiv.org/abs/1711.05101. arXiv:1711.05101.

[33] L. Gao, J. Tow, B. Abbasi, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, A. Le Noac'h, H. Li, K. McDonell, N. Muennighoff, C. Ociepa, J. Phang, L. Reynolds, H. Schoelkopf, A. Skowron, L. Sutawika, E. Tang, A. Thite, B. Wang, K. Wang, A. Zou, The language model evaluation harness, 2024. URL: https://zenodo.org/records/12608602. doi:10.5281/zenodo.12608602.

[34] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, J. Schulman, Training verifiers to solve math word problems, 2021. URL: https://arxiv.org/abs/2110.14168. arXiv:2110.14168.

[35] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, J. Steinhardt, Measuring mathematical problem solving with the math dataset, NeurIPS (2021).

[36] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, Y. Wu, B. Neyshabur, G. Gur-Ari, V. Misra, Solving quantitative reasoning problems with language models, 2022. arXiv:arXiv:2206.14858.

[37] M. Polignano, P. Basile, G. Semeraro, Advanced natural-based interaction for the italian language: Llamantino-3-anita, 2024. URL: https://arxiv.org/abs/2405.07101. arXiv:2405.07101.

[38] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, 2023. URL: https://arxiv.org/abs/2310.06825. arXiv:2310.06825.

[39] A. Y. et al., Qwen3 technical report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

[40] S. Banerjee, A. Lavie, METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, in: J. Goldstein, A. Lavie, C.-Y. Lin, C. Voss (Eds.), Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Association for Computational Linguistics, Ann Arbor, Michigan, 2005, pp. 65–72. URL: https://aclanthology.org/W05-0909/.

[41] T. Kočiský, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, E. Grefenstette, The NarrativeQA reading comprehension challenge, Transactions of the Association for Computational Linguistics 6 (2018) 317–328. URL: https://aclanthology.org/Q18-1023/. doi:10.1162/tacl_a_00023.

[42] A. Scirè, S. Conia, S. Ciciliano, R. Navigli, Echoes from alexandria: A large resource for multilingual book summarization, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 853–867. URL: https://aclanthology.org/2023.findings-acl.54/. doi:10.18653/v1/2023.findings-acl.54.

[43] Y. Bai, X. Lv, J. Zhang, H. Lyu, J. Tang, Z. Huang, Z. Du, X. Liu, A. Zeng, L. Hou, Y. Dong, J. Tang, J. Li, LongBench: A bilingual, multitask benchmark for long context understanding, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 3119–3137. URL: https://aclanthology.org/2024.acl-long.172/. doi:10.18653/v1/2024.acl-long.172.

A. Timing and CO_2 Emissions Estimates

To quantify both the computational effort and environmental footprint of our training end experiments we compute energy and CO_2 estimates assuming: Average GPU power draw: 300 W under full load. Data-center PUE (power usage effectiveness): 1.2. Grid emission factor: 0.28 kg CO_2/kWh (typical for the European grid).

Total energy consumed per GPU-hour is

$$E_{\mathrm{kWh/GPUh}} = 0.3\ \mathrm{kW} \times 1.2 = 0.36\ \mathrm{kWh/GPUh},$$

and CO_2 emitted per GPU-hour is

$$\begin{aligned} M_{\mathrm{CO_2/GPUh}} &= 0.36\ \mathrm{kWh} \times 0.28\ \frac{\mathrm{kg}}{\mathrm{kWh}} \\ &\approx 0.10\ \mathrm{kg\ CO_2/GPUh}. \end{aligned}$$

We estimate that the continual training of four recipes, Recipe 1 (3.5 days) and Recipes 2, 3, and 3_{16k} (7 days each), on 64 GPUs corresponds to a total GPU-time of $\approx 37\,632$ GPUh.
Using an emission factor of $0.10\ \mathrm{kg\,CO_2/GPUh}$, this yields about $3.8\ \mathrm{t\ CO_2}$.
With respect to the instruction tuning process, considering the same number of GPUs,the standard 4 096-token variant required approximately 3000 GPU-hours, emitting roughly 3 t CO_2. The long-context 16 384-token variant ran for about double the time (6000 GPU-hours), producing approximately 6 tons of CO_2.

B. Data Processing

This Section outlines the data processing steps applied to the various datasets used in the three main recipes described in Section 3.1.

Benchmarks. We utilized the translated benchmarks from ITA-Bench [17], specifically leveraging the training sets (when available) from both the original and translated versions. We formatted these through defined prompts consistent with `LM-Evaluation-Harness` [33].

Wikisource. We downloaded the Hugging Face version of the Wikisource dataset, available at: https://huggingface.co/datasets/wikimedia/wikisource.

Gazzetta Ufficiale. We downloaded the Hugging Face version of the Gazzetta Ufficiale dataset, available at: https://huggingface.co/datasets/mii-llm/gazzetta-ufficiale.

Wikipedia. For Recipe-1, we used the Hugging Face version of the Wikipedia dataset, available at: https://huggingface.co/datasets/wikimedia/wikipedia. While for Repice 2 and 3 we used an updated version collected and processed by us with pages created up to 2024.

RedPajama. We retrieved the RedPajama dataset from Hugging Face: https://huggingface.co/datasets/togethercomputer/RedPajama-Data-V2. We performed deduplication using the provided metadata and extracted the text from the 'head' partition of each dump. For Recipe-1, we used the 2023-14 dump, while for Recipes 2 and 3 we additionally used dumps 2023-06, 2022-49, and 2022-40. We filtered out texts with fewer than 500 words.

Gutenberg. We collected texts from Project Gutenberg via Hugging Face: https://huggingface.co/datasets/manu/project_gutenberg.

Fineweb-Edu. We used the Fineweb-Edu dataset from Hugging Face: https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu, specifically the `sample-100BT` branch. This is a random subset of the full dataset. For Recipe-1, we selected pages with a minimum quality score of 3.8; for Recipes 2 and 3, we applied a threshold of 4.0.

Dolmino. The Dolmino data, specifically the `math` and `stackexchange` subsets, were obtained from: https://huggingface.co/datasets/allenai/dolmino-mix-1124.

FLAN. We downloaded the FLAN dataset from https://huggingface.co/datasets/allenai/dolma. We selected only the examples using the following prompt formats: `fs_opt`, `fs_noopt`, `zs_opt`, and `zs_noopt`.

The Stack. We collected data from the Stack dataset at: https://huggingface.co/datasets/bigcode/the-stack-v2-train-smol-ids. We included only code samples from the `refs/heads/master` and `refs/heads/main` branches, and further filtered to include only repositories with at least 10 GitHub stars.

Books3. We used a previously obtained copy of the Books3 dataset, which is no longer publicly available for download.

C. INDAQA

In this Section, we present additional details on the dataset we built, INDAQA. We retain samples asking the same questions with different formulations, following the approach in NarrativeQA. This design choice preserves valuable linguistic variation that may prove instrumental for future analyses examining the effects of question reformulation on QA system performance. While we maintain paraphrased questions, we eliminate exact duplicates from the dataset, ensuring that each unique reference answer is preserved only once.

We present some of the discarded questions in Table 11. These samples were filtered using several RegEx. We refined the RegEx patterns by manually validating their impact on a subset of 17 documents (646 QA samples).

Finally, we also show the prompts used to generate these samples in Tables 12. To ensure uniqueness, all QA pairs for each book were generated in a single in-

Question type	Transl.	Count	%
Cosa	*What*	4309	31.5
Chi	*Who*	3517	25.7
Quale/i	*Which*	2496	18.2
Come/In che modo	*How*	1496	10.9
Dove	*Where*	1105	8.1
Perché	*Why*	413	3.0
Quanto/a/i/e	*How much*	146	1.1
Quando	*When*	29	0.2
MISCELLANEA	*OTHER*	185	1.4

Table 10
Statistics on the type of questions in INDAQA. The majority of questions asks about events (*What*) and characters (*Who*). Due to the short summary length, models struggled to generate *Why* and *Where* question.

ference step and were later deduplicated. This process was repeated three times with different answer length requirements.

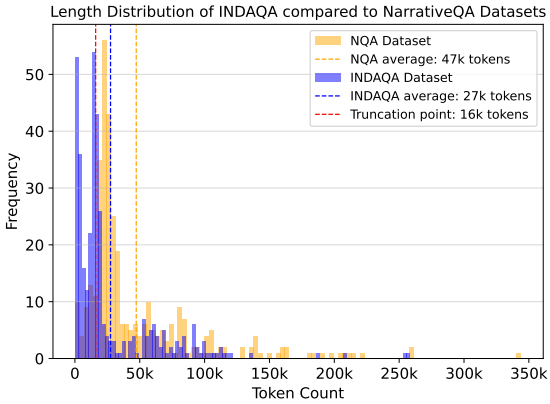


Figure 2: Histogram showing the differences between our dataset, INDAQA, and the test set of NarrativeQA (NQA).

D. ITALIC-GEN

This Section provides additional details on the adaptation of the ITALIC dataset [19] from a multiple-choice format to a *free-form* generative QA setting. Such adaptations must extend beyond simply extracting correct answers from the provided options, requiring systematic analysis of the underlying sample characteristics and question types.

The original ITALIC dataset contains 10,000 instances divided into two primary categories: Language Capability and Culture and Commonsense. Due to the heterogeneous nature of the underlying data sources, not all samples adhere to the standard question format. Specifically:

1. Many instances follow a *sentence completion* style, where the correct completion has to be selected from the multiple options.
2. Additionally, certain samples depend on contextual information that is embedded within the answer choices themselves, making the removal of options infeasible without compromising the question quality.
3. Finally, some questions, while not strictly requiring all four options to be answerable, become insufficiently specific without the provided choices, potentially leading to ambiguous interpretations.

Moreover, the last two cases mostly require the model to reproduce *verbatim* one of the choices, which is significantly different from the open-ended QA task.

After automatic and manual inspection, we found that the majority of samples in the Language Capability category suffer from these structural limitations, with many instances exhibiting multiple concurrent issues, resulting in the need for heavy modifications to be adopted. While such characteristics are appropriate for multiple-choice QA frameworks, they present significant challenges for generative QA tasks. Consequently, we excluded all Language Capability samples from our experiments, resulting in ITALIC-GEN containing exclusively instances from the Culture and Commonsense category.

We set up a pipeline to check and modify the remaining samples to ensure compatibility with the generative QA setting. First, we employ Gemini-2.0-Flash to reformat statements not ending with a question mark (?) into proper interrogative form, standardizing the format across all instances (issue number 1). We also require the LLM to ensure proper coordination between question and answer. Manual verification of the results identified three instances that required correction where automatic reformatting failed to produce valid questions.

Then, we filter the samples that would become unanswerable without access to the multiple-choice options (issue number 2) by first using a set of RegEx (both on questions and correct choices), and then employing the LLM to classify samples based on the context provided in the question alone. We applied this validation process to the whole dataset, both original and reformatted samples. During the initial inspection of the samples, we noted that the third issue predominantly affects samples in the Language Capability category. Since ITALIC-GEN exclusively comprises Culture and Commonsense samples, we did not implement additional filtering based on this criterion. We do acknowledge that some instances in ITALIC-GEN may present significant challenges for current generative QA systems.

Error type	Question	Answers
Unanswerable	I corteggiatori sono rivali tra loro?	1) Non è specificato. 2) Il testo non lo dice.
Unanswerable	Cosa prova il Conte nei confronti del letterato?	1) Disprezzo. 2) Il testo non specifica i sentimenti.
Meta	Cosa descrive ciascun capitolo?	1) Cronache. 2) Riassunti di cronache.
Meta	Qual è il titolo del testo?	1) Il titolo non è specificato. 2) Non c'è alcun titolo.

Table 11
Types of samples in INDAQA filtered by our pipeline. We remove the samples even if one of the reference answers is acceptable.

System Prompt

```
Sei un esperto di letteratura.
Il tuo compito è quello di generare domande e risposte sulla trama di un testo letterario.
```

User Prompt

```
TESTO: {summary}
Genera 20 domande diverse relative alla trama del testo.
Per ogni domanda, genera due possibili risposte, entrambe corrette e complete.
Le domande devono essere chiare e non ambigue; se il testo è breve, genera comunque 20
domande.
Entrambe le risposte devono essere brevi (max 5 parole), complete e rispecchiare fedelmente
il testo originale.
Le risposte possono anche essere quasi identiche.
Segui il formato, non aggiungere altro:
Domanda: <domanda>
Risposta A: <risposta>
Risposta B: <risposta>
```

Table 12
Prompts used to generate the QA samples for the INDAQA dataset. We used Gemini-2.0-Flash and Gemini-2.0-Flash-Lite as our Generators.

Issue	Question	Choices
1	"The Young Pope" è il titolo della serie ideata e diretta da:	1) Kim Rossi Stuart 2) Christian De Sica 3) Roberto Benigni **4) Paolo Sorrentino**
2	Con l'espressione "Schiaffo di Anagni" si è soliti indicare:	1) Lo schiaffo che Anagni diede a papa Bonifacio VIII 2) L'offesa che Bonifacio VIII recò ad Anagni **3) L'oltraggio che subì papa Bonifacio VIII ad Anagni** 4) -
2	Quale frase contiene un complemento di compagnia?	1) La ballerina aspettava con ansia il giorno del suo debutto **2) Sono andato al lago con mia sorella per prendere il sole** 3) Il medico garantisce che con questa crema passerà il rossore 4) Con questa velocità non riuscirai mai a finire il lavoro per domani
3	La frase "Sono felice" contiene:	1) un complemento oggetto 2) un complemento indiretto 3) **un predicato nominale** D) un predicato verbale

Table 13
Instances of ITALIC that cannot be used in a generative QA setting. While we can keep the first two instances, after proper modifications, the last two neessarily require the options as context.