Minerva LLMs: The First Family of Large Language Models Trained from Scratch on Italian Data

Riccardo Orlando[1,†], Luca Moroni[1,†], Pere-Lluís Huguet Cabot[1,†], Edoardo Barba[1], Simone Conia[1], Sergio Orlandini[2], Giuseppe Fiameni[3] and Roberto Navigli[1,*]

[1]Sapienza NLP Group, Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza University of Rome, Italy
[2]CINECA, Bologna, Italy
[3]NVIDIA, Santa Clara, California, USA

Abstract

The growing interest in Large Language Models (LLMs) has accelerated research efforts to adapt these models for various languages. Despite this, pretraining LLMs from scratch for non-English languages remains underexplored. This is the case for Italian, where no truly open-source research has investigated the pretraining process. To address this gap, we introduce Minerva (https://nlp.uniroma1.it/minerva), the first family of LLMs trained entirely from scratch on native Italian texts. Our work is the first investigation into the challenges and opportunities of pretraining LLMs specifically for the Italian language, offering insights into vocabulary design, data composition, and model development. With Minerva, we demonstrate that building an LLM tailored to a specific language yields numerous practical benefits over adapting existing multilingual models, including greater control over the model's vocabulary and the composition of its training data. We provide an overview of the design choices, pretraining methods, and evaluation metrics used to develop Minerva, which shows promising performance on Italian benchmarks and downstream tasks. Moreover, we share the lessons learned throughout Minerva's development to support the academic and industrial communities in advancing non-English LLM research. We believe that Minerva serves as an important step towards closing the gap in high-quality, open-source LLMs for non-English languages.

Keywords

Large Language Models, Language Modeling, Italian Language, LLM Pretraining

1. Introduction

Large Language Models (LLMs) have revolutionized the way Natural Language Processing (NLP) tasks are approached, achieving remarkable results in existing areas and opening the door to entirely new research directions and applications. As a result, the energy and resources dedicated to the study and creation of LLMs are growing exponentially. However, most LLMs – both closed- and open-source – are predominantly designed for English, posing significant challenges and limitations for their use in non-English settings. In practice, generating Italian text using multilingual or language-adapted English models, e.g., from Mistral [1] or Llama [2, 3], is computationally more expensive and often less effective compared to using a model specifically designed for the Italian language. This inefficiency stems from the vocabulary of an English or multilingual LLM – i.e., the lexical units, or tokens, that the model can use to compose text – when it is not optimized for the Italian language, resulting in Italian words being split into an excessive number of tokens. Consequently, this creates longer sequences of tokens, slower generation times, and higher computational costs, especially since many popular attention mechanisms have a quadratic complexity with respect to sequence length.

Efforts to create language-specific LLMs are increasing, and fall primarily into two main categories: i) adapting existing English-centric LLMs to other languages, and ii) training LLMs from scratch. The advantages of adapting existing English-centric LLMs to other languages are enticing: starting with a proven model can reduce the computational requirements, and adaptation can be achieved with relatively modest amounts of data. There are several language adaptation techniques, which range from fine-tuning the model on data for the target language [4, 5] to modifying the model's architecture [6, 7, 8], making these techniques flexible for different budgets and objectives. However, these techniques may not fully capture language-specific nuances and can degrade the performance in the original language, indeed an undesirable effect. Alternatively, training LLMs from scratch provides the freedom to make design choices tailored to the linguistic features of the target language—including morphology, lexicon, syntax, and semantics—which are often overlooked in English-centric models [9]. It

CLiC-it 2024: Tenth Italian Conference on Computational Linguistics, Dec 04 — 06, 2024, Pisa, Italy
*Corresponding author.
†These authors contributed equally.
orlando@diag.uniroma1.it (R. Orlando); moroni@diag.uniroma1.it (L. Moroni); huguetcabot@diag.uniroma1.it (P. H. Cabot); barba@diag.uniroma1.it (E. Barba); conia@diag.uniroma1.it (S. Conia); s.orlandini@cineca.it (S. Orlandini); gfiameni@nvidia.com (G. Fiameni); navigli@diag.uniroma1.it (R. Navigli)

also allows for incorporating culturally relevant content, reducing biases that might be present in models primarily trained on English data, thus leading to more inclusive and accurate representations of language use. Unfortunately, while there are several efforts on adapting English-centric LLMs to the Italian language, e.g., Llamantino-2 [4], Llamantino-3 [5], DanteLLM [10], and Camoscio [11], *inter alia*, there is no truly open-source endeavor exploring what can be achieved by training an LLM from scratch on Italian data.

With this work, we follow the latter path and introduce Minerva, the first family of LLMs designed specifically for the Italian language and pretrained on Italian text.[1] We present the design choices for our models, our data processing, and the evaluation results regarding our Minerva LLMs, showing that our models – with 350M, 1B, 3B, and 7B parameters – outperform comparable multilingual models and even rival larger models adapted for Italian. We conclude with a discussion on the benefits and challenges of pretraining LLMs from scratch for the Italian language, sharing our experience and findings to provide valuable insights for the academic and industrial communities interested in training non-English LLMs from scratch. Lastly, we describe the technical details of Minerva-7B, our latest model with 7.4 billion parameters, for which we share our initial results.

2. Building a Pretraining Dataset for Italian LLMs

The field of LLMs is growing at an astonishing pace, with new models, datasets, benchmarks, and techniques presented every week. However, over the past few months, academic and industrial researchers have increasingly recognized the fundamental role of the data used to pretrain LLMs. Unsurprisingly, the majority of the leading companies are not releasing their training data as they seek to maintain an advantage over the competition, with very few exceptions (e.g. OLMo by AllenAI [12] and OpenELM by Apple [13]). In this section, we describe the different sources of data used in the training of the Minerva models, and Table 1 provides an overview of these (cf. Appendix A for more details). Most importantly, the training datasets we used are entirely available online, making our process transparent and allowing researchers to better study the connection between pretraining data and model behavior.

2.1. Data Sources

The training data for our Minerva models consists of three main categories: Italian, English, and code data.

[1] https://nlp.uniroma1.it/minerva

Dataset		Minerva – Model Size			
Name	**Lang.**	**350M**	**1B**	**3B**	**7B**
RedPajama-V2	Italian	–	–	–	894B
CulturaX	Italian	35B	100B	330B	237B
Wikipedia	Italian	–	–	–	1.3B
Gutenberg	Italian	–	–	–	0.15B
Wikisource	Italian	–	–	–	0.12B
EurLex	Italian	–	–	–	1.6B
Gazzetta Ufficiale	Italian	–	–	–	1.7B
FineWeb	English	–	–	–	1,076B
CulturaX	English	35B	100B	330B	–
Wikipedia	English	–	–	–	5.3B
ArXiv	English	–	–	–	33B
Gutenberg	English	–	–	–	7B
StackExchange	English	–	–	–	22B
The Stack V2	Code	–	–	–	201B
Total # of tokens		**70B**	**200B**	**660B**	**2.48T**

Table 1
Datasets used to train Minerva with their languages (second column) and number of tokens (third to sixth columns).

We only use the code data to train our largest model, i.e., Minerva-7B.

2.1.1. Italian Data

Web data. The majority of the text used to train LLMs is sourced from Web-scraped data, typically from CommonCrawl (CC). Therefore, a significant portion of Italian text included in our training datasets is also of this nature, inherently exposing our models to potential biases and toxic content commonly found on the Web. Because preprocessing techniques, such as language identification, perplexity filtering, deduplication, and content classification are computationally expensive, the most sensible choice is thus to rely on preprocessed collections, such as CulturaX [14] and RedPajama v2 [15]. These collections already include Italian data, and have undergone various levels of filtering and deduplication, as discussed in Section 2.2.

Curated data. While Penedo et al. [16] suggest that high-quality Web data is sufficient on its own to train LLMs, curated data sources are often used to further improve the model performance and introduce a broader diversity of data types, such as encyclopedic and academic text [17], as well as scientific and math-related text. Therefore, we include curated texts from several sources, including Wikipedia (encyclopedic/world knowledge data), EurLex and Gazzetta Ufficiale (law, economics, and politics), and the Gutenberg Project (novels, poetry, etc.).

2.1.2. English Data

Web data. Mirroring our approach with the Italian data, we use preprocessed collections of English data

from the Web. Given that English is the most popular language on the Internet and has been the primary focus of LLM research, there are numerous options that already provide a large amount of tokens from filtered, deduplicated, and cleaned sources. For our Minerva-350M, 1B, and 3B models, we collect data from the English partition of CulturaX, capping the number of tokens to the same amount as the Italian ones, as shown in Table 1. Instead, to train Minerva-7B, we use a portion of FineWeb [18], which includes filtered and deduplicated CC dumps with various timestamps. Specifically, we use the CC dumps from 2023-14 to 2024-18 to match the total number of tokens in the Italian Web partition of our training data.

Curated sources. We include the 5.3B tokens from the English Wikipedia and 7B tokens from the copyright-free books in Project Gutenberg. Additionally, we include data from arXiv and StackExchange, which are included in the RedPajama dataset.

2.1.3. Code Data

Previous work has highlighted the importance of including source code in the pretraining corpus of an LLM, in order to improve not only its code understanding and generation, but also its general reasoning capabilities [19] even for tasks that do not directly involve or require programming. Therefore, for our largest model – Minerva-7B – we also include a portion of code data. More specifically, we extract 200B tokens from The Stack V2 [20], selecting the data from their deduplicated partition, which includes 17 of the most popular programming languages on GitHub.

2.2. Data Preprocessing

As mentioned above, our preprocessing effort remains minimal, as we rely on the preprocessing pipelines used in CulturaX, RedPajama, and FineWeb. To evaluate the content and quality of our training data, we employ the methodology described in Elazar et al. [21] to analyze the URL domain distribution within the Italian partition of CulturaX and RedPajama, as these partitions had never been utilized in training an LLM prior to Minerva. We provide an overview of our analysis together with a few insights in Appendix B.

2.3. Data Filtering and Deduplication

Previous work on English-centric LLMs [22] has already emphasized the importance of training LLMs on "clean" data. Two of the most important parts of data cleaning are filtering, i.e., removing content that does not satisfy a set of criteria, and deduplication, i.e., removing portions of text that appear too often so as to minimize memorization.

As mentioned above, for the corpus used to train the Minerva models, we rely mainly on collections of data that has already been filtered and deduplicated. However, there are some minor considerations that depend on each collection of data. More specifically, we use CulturaX as-is, relying on their filtering and deduplication pipeline. Unfortunately, RedPajama v2 is not filtered and deduplicated; however, its data is tagged with meta-information that can be used to apply filtering and deduplication. Such metadata includes, for example, the perplexity score of each text computed via a language model trained on Wikipedia, which is used to partition RedPajama v2 into three partitions: *head*, *middle*, *tail*. For our training corpus, we only include a document if it is classified as *head* or *middle* according to its perplexity score. Moreover, we use the precomputed metadata to remove exact duplicates and apply fuzzy deduplication. The latter is performed by using the hash provided for each document with Locality Sensitive Hashing and Jaccard similarity 0.7 to decide whether two documents are fuzzy duplicates. Note that we only apply fuzzy deduplication within each CC dump, rather than across all the dumps. This decision is motivated by two observations: first, applying fuzzy deduplication across all CC dumps is computationally expensive; second, previous work [18] has shown that per-CC deduplication is not only sufficient, but is also beneficial, when training English LLMs.

3. Minerva LLMs

In this section, we provide an overview of the Minerva LLMs: we describe their tokenizers, the design choices behind the model architecture, and how we trained the resulting LLMs.

3.1. Vocabulary and Tokenizers

The vocabulary of an LLM is mainly impacted by its size, i.e., the number of tokens in the vocabulary itself, and how the tokenizer is trained, i.e., which tokens make up the vocabulary. These two factors impact the fertility of the resulting tokenizer, which measures the average number of tokens (subwords) into which a word is split. Tokenizers with lower fertility are preferable, as the input and output sequences they produce are shorter, resulting in an efficiency gain, especially as most attention mechanisms are quadratic with respect to the sequence length. Unsurprisingly, the vocabulary allocation of an English-centric LLM minimizes the fertility of English text, and results in high fertility values for Italian text, as shown in Table 2.

		Fertility (↓ – *lower is better*)			
		CulturaX		*Wikipedia*	
Tokenizer	**|Vocab|**	*Ita*	*Eng*	*Ita*	*Eng*
Mistral-7B	32,000	1.87	1.32	2.05	1.57
Gemma-7B	256,000	1.42	1.18	1.56	1.34
Minerva-350M	32,768	1.39	1.32	1.66	1.59
Minerva-1B	32,768	1.39	1.32	1.66	1.59
Minerva-3B	32,768	1.39	1.32	1.66	1.59
Minerva-7B	51,200	1.32	1.26	1.56	1.51

Table 2
Fertility rates (lower is better) for Minerva tokenizers compared to other LLMs. The fertility rates are computed on a randomly sampled collection of texts from CulturaX and Wikipedia in both Italian (Ita) and English (Eng).

Given the importance for our Minerva LLMs of having a low fertility on Italian text, we intentionally train the Minerva tokenizer on a balanced mix of English and Italian data (and code data for the 7B model). Our analysis shows that this strategy leads to a much improved fertility on Italian data, while at the same time maintaining similar fertility on English data. More specifically, for Minerva-350M/1B/3B, we opted for a vocabulary size similar to that of Mistral-7B (around 32k tokens): in this case, the fertility of the Minerva tokenizer is ~20% better than the Mistral tokenizer on the Italian Wikipedia and only ~1% worse on the English Wikipedia. Following recent trends in LLMs, for Minerva-7B, we increased the vocabulary size to around 50k tokens, which resulted in a further fertility improvement of ~6% and ~5% on the Italian and English Wikipedias, respectively, notwithstanding the addition of code data to the training data. We provide more details on the tokenizer in Appendix C.

3.2. Model Architecture

While the field of LLMs is moving rapidly, one of the best models when our efforts started was Mistral. Therefore, our Minerva LLMs are based on Mistral's model architecture. The Minerva LLMs are, therefore, a family of decoder-only transformer models, with a few standout features, such as grouped-query attention (GQA) [23], which boosts inference speed and reduces memory requirements for increased throughput, and sliding window attention (SWA) [24, 25], which manages longer sequences more efficiently at reduced computational costs. Specifically, the GQA is configured to share one key-value pair every four queries, while the SWA configuration handles up to 2,048 tokens with a maximum context length of 16,384 tokens. We build four models with different sizes by scaling the number of attention heads, hidden size, intermediate size, and hidden layers, while maintaining a ratio of ~3.5 between the hidden size and intermediate size, as in the original Mistral model. However, following the more recent model releases by Mistral, Minerva-7B does not use SWA. Instead, it implements full attention across its entire context length, which can extend up to 4096 tokens, i.e., double the number of tokens for the SWA used in Minerva-350M/1B/3B. The parameters for each model size are detailed in Table 3, for which we provide a more in-depth description in Appendix D.

Building Minerva on top of Mistral's model architecture also brings other benefits, such as broad compatibility with the ecosystem of libraries, frameworks, and tools that has emerged over recent months, including llama.cpp [26], FlashAttention [27], and vLLM [28].

3.3. Model Training

We train all the Minerva LLMs using MosaicML's LLM Foundry.[2] The training process is conducted on the Leonardo Supercomputer[3] hosted and maintained by CINECA. Each node in Leonardo is equipped with 4 × custom NVIDIA A100 SXM4 with 64GB of VRAM.

All our models are trained using the AdamW optimizer [29] with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $eps = 10^{-8}$ (with the only exception being Minerva-7B, which is trained using $eps = 10^{-5}$) on a standard causal language modeling training objective. To smooth the training process, we follow standard practice in the literature and employ a warmup-then-cooldown learning rate scheduling. More specifically, we first increase the learning rate linearly during the initial training phase (2% of the total number of training steps for Minerva-350M/1B/3B and 0.3% for Minerva-7B) until the peak learning rate is reached (2×10^{-4} for Minerva-350M/1B/3B, 3×10^{-4} for Minerva-7B), and then decrease the learning rate with a cosine scheduling until the end of the training process. The hyperparameters used for each model are shown in Table 7.

4. Evaluation

We measure the 0-shot performance of our Minerva LLMs on ITA-Bench [30], a suite of benchmarks that have been created either by translating existing benchmarks from other languages, or by adapting existing Italian benchmarks so that they can be used for LLM evaluation. ITA-Bench includes a set of 10 benchmarks commonly used to evaluate LLMs, namely, ARC Challenge (ARC-C), ARC Easy (ARC-E) [31], BoolQ [32], GSM8K [33], HellaSwag (HS) [34], MMLU [35], PIQA [36], SciQ [37], TruthfulQA [38], and Winogrande (WG) [39]. Overall, these benchmarks offer a comprehensive view of the capabilities of an LLM on a wide variety of aspects, including scientific knowledge, world knowledge (e.g., geography, politics, economics), commonsense knowledge, physical

[2] https://github.com/mosaicml/llm-foundry
[3] https://leonardo-supercomputer.cineca.eu/

Model	Params	Layers	Hidden Size	Inter. Size	Att. Heads	KV Heads	SW Length	Ctx. Length
Minerva-350M	352M	16	1152	4032	16	4	2048	16,384
Minerva-1B	1.01B	16	2048	7168	16	4	2048	16,384
Minerva-3B	2.89B	32	2560	8960	32	8	2048	16,384
Minerva-7B	7.40B	32	4096	14336	32	8	None	4,096

Table 3
Overview of the main hyperparameters for our Minerva models. We include the number of parameters (approximately, 350M, 1B, 3B, and 7B) and the corresponding number of layers, hidden size, intermediate size, attention heads, key-value heads, sliding window length, and maximum context length.

Size	Name	ARC-C	ARC-E	BoolQ	GSM8K	HS	MMLU	PIQA	SciQ	TQA	WG	AVG
0.4B	Minerva-350M-base-v1.0	24.6	36.4	60.7	48.2	32.6	25.7	59.5	63.7	46.5	58.4	45.6
1B	Minerva-1B-base-v1.0	26.6	42.2	57.1	49.7	39.6	27.0	62.9	73.5	44.6	60.0	48.3
3B	OpenELM-3B	27.0	37.9	60.9	49.7	40.7	28.3	56.7	81.8	47.3	58.4	48.9
3B	XGLM-2.9B	27.5	41.4	59.1	65.7	44.5	27.4	59.9	77.8	43.1	60.2	50.6
3B	Minerva-3B-base-v1.0	31.4	49.1	62.1	55.8	52.9	29.2	66.9	79.9	41.4	62.2	53.1
7B	OLMo-7B-0724-hf	30.7	44.0	72.9	52.5	47.9	30.9	58.7	85.1	44.6	61.2	52.8
7B	LLaMAntino-2-7b	33.7	50.8	70.9	52.2	54.9	33.8	64.4	86.1	44.3	64.1	55.5
7B	Minerva-7B-base-v1.0	38.4	57.7	68.2	52.2	60.4	34.0	69.4	85.2	42.5	63.9	57.2
7B	Mistral-7B-v0.1	42.8	61.3	78.2	56.1	60.4	38.0	65.5	90.8	43.5	68.8	60.5
8B	Llama-3.1-8B	44.0	61.1	78.0	57.8	62.9	38.7	67.7	90.3	43.0	69.2	61.3

Table 4
Zero-shot evaluation results of the Minerva models on a set of standard benchmarks translated from English to Italian.

interactions, coreference, and math reasoning, among others. Employing automatically-translated benchmarks is far from ideal, but it allows us to better compare the scores obtained in Italian with those obtained in English, while awaiting as the Italian research community develops Italian-specific benchmarks [40].

As shown in Table 4, the average performance of the Minerva models increases steadily with the model size. For our 3B model, we also provide a comparison with two models of the same size: XGLM [41], a multilingual LLM by META, and OpenELM [42], a very recent English-only model developed by Apple. Our evaluation shows that Minerva-3B outperforms XGLM and OpenELM by a significant margin, i.e., +4.4% and +3.7% on average.

Finally, Minerva-7B achieves the highest performance among the Minerva LLMs family, as expected. Notably, Minerva-7B, achieves a higher average score than Llamantino-2. This is an interesting comparison because the pretraining data for Llama-2, i.e., the pretrained LLM used to build Llamantino-2, is not available and has never been disclosed, making the model open-weights but not entirely open-source.[4] When compared to closed-sourced LLMs such as Mistral-7B-v0.1 or Llama-3.1-8B, Minerva still lags behind in some tasks, such as BoolQ or GSM8K, which may require better reasoning capabilities and/or more pretraining data. As we can observe from Figure 1, which tracks the progress of Minerva-7B on ITA-Bench every 10,000 training steps, the model is still slowly improving towards the end of the pretraining phase, suggesting that a larger training corpus or multiple epochs may be beneficial in future developments.

[4]We stress that, for Llamantino-2, only the data that has been used for the language adaptation process is available, whereas the pretraining data is not.

5. Downstream tasks

In this section, we show the results of the Minerva models when adapted to two downstream applications. This analysis is particularly relevant for Minerva-350M and Minerva-1B, which can be utilized for specific tasks rather than as general-purpose models, offering lower computational costs. The tasks in this analysis include: i) Italian Abstractive News Summarization, and ii) Machine Translation, in both directions (IT-EN and EN-IT).

News Summarization. Following Sarti and Nissim [43], we fine-tune Minerva models (up to 3B) on a concatenation of two Italian news summarization datasets: Fanpage.it and Il Post newspapers [44]. A detailed overview of the hyperparameters used to train our models is provided in Appendix E. We can find that Minerva-3B obtains the best results (0.30 vs 0.29 of the second best in terms of Rouge-L); however, it is not as parameter-efficient as IT5-Large, probably because encoder-decoder models are more suitable for fine-tuning than decoder-only models [45]. In Table 8, we report the full results of Minerva fine-tuned on the aforementioned datasets and compared to baselines in Sarti and Nissim [43], which

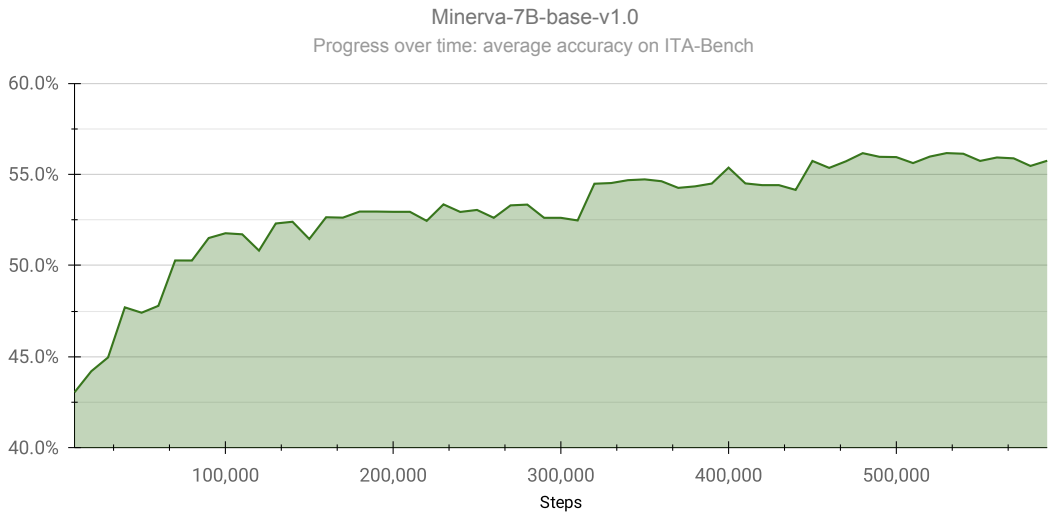

Figure 1: Tracking the progress of Minerva-7B during its pretraining process. Here, we report the average accuracy on ITA-Bench every 10,000 steps, i.e., every 40B tokens approximately.

include mBART, mT5, and IT5.

Machine Translation. We also evaluate our Minerva LLMs in few-shot [46] machine translation on two benchmarks, FLORES [47] and OPUS-100 [48]. We explore how LLMs perform this task relying only on in-context-learning few-shot examples, reporting our results with 5-shot prompting. We rely on the vLLM library [28] and change the default parameters with temperature=0 and max_tokens=512.

We highlight that Minerva-3B reaches competitive results in MT in both EN-IT (84.8 on Flores and 76.7 on Opus in terms of COMET score) and IT-EN (85.7 and 78.0). Compared with other models of similar size, Minerva-3B shows strong results when the target language is Italian (+1.7 and +2.7 compared to Gemma-2B and Qwen-1.5B on Opus). Minerva-7B further showcases this by achieving the highest performance among models tested when translating from English into Italian. The full results are reported in Table 5.

	FLORES		OPUS	
Model	**EN-IT ↑**	**IT-EN ↑**	**EN-IT ↑**	**IT-EN ↑**
Minerva-1B	66.37	73.72	57.40	64.61
Minerva-3B	84.83	85.67	76.74	78.04
Minerva-7B	**87.02**	87.20	**79.07**	79.91
Gemma-2B	83.31	86.51	75.05	78.94
Qwen-1.5B	80.18	86.16	74.01	78.95
TinyLlama-1.1B-v1.1	73.40	83.62	65.72	75.44
LLaMa-2-7B	85.24	87.47	77.30	80.36
Mistral-7B	86.56	**87.75**	78.08	80.56
Qwen-7B	86.00	87.66	78.50	**81.21**

Table 5
COMET scores measure the translation capabilities of our Minerva models and other LLMs on the FLORES and OPUS datasets. This evaluation is conducted in a 5-shot setting, where each model receives five random translation examples from the development set before the test instance.

6. Conclusion and Future Work

In this paper, we demonstrated the feasibility and benefits of pretraining Italian language models from scratch, which not only improves the computational efficiency and performance of an LLM for a target language but reduce linguistic biases inherited from English training corpora [49]. The Minerva models (https://nlp.uniroma1.it/minerva) showcase promising results on a variety of Italian benchmarks and downstream tasks, including news summarization and machine translation. Most importantly, we describe, for the first time, the process of creating an Italian pretraining corpus with more than 1T tokens, and we share findings and insights into the pretraining process of Italian LLMs with the academic and industrial communities, paving the way for future research in training non-English language models. We hope that our contributions will represent a stepping stone for future work on language-specific and multilingual large-scale language modeling.

Acknowledgments

Pere-Lluís Huguet Cabot, Simone Conia and Edoardo Barba are fully funded by the PNRR MUR project PE0000013-FAIR. Roberto Navigli also acknowledges the support of the PNRR MUR project PE0000013-FAIR. The authors acknowledge the CINECA award IsB28_medit under the ISCRA initiative for the availability of high-performance computing resources and support.

References

[1] 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.

[2] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. URL: https://arxiv.org/abs/2302.13971. arXiv:2302.13971.

[3] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open foundation and fine-tuned chat models, 2023. URL: https://arxiv.org/abs/2307.09288. arXiv:2307.09288.

[4] 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.

[5] M. Polignano, P. Basile, G. Semeraro, Advanced natural-based interaction for the italian language: Llamantino-3-anita, 2024. arXiv:2405.07101.

[6] M. Ostendorff, G. Rehm, Efficient language model training through cross-lingual and progressive transfer learning, arXiv preprint arXiv:2301.09626 (2023).

[7] K. Dobler, G. de Melo, FOCUS: Effective embedding initialization for monolingual specialization of multilingual models, 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. 13440–13454. URL: https://aclanthology.org/2023.emnlp-main.829. doi:10.18653/v1/2023.emnlp-main.829.

[8] Z. Csaki, B. Li, J. Li, Q. Xu, P. Pawakapan, L. Zhang, Y. Du, H. Zhao, C. Hu, U. Thakker, Sambalingo: Teaching large language models new languages, arXiv preprint arXiv:2404.05829 (2024).

[9] M. Faysse, P. Fernandes, N. Guerreiro, A. Loison, D. Alves, C. Corro, N. Boizard, J. Alves, R. Rei, P. Martins, et al., Croissantllm: A truly bilingual french-english language model, arXiv preprint arXiv:2402.00786 (2024).

[10] A. Bacciu, C. Campagnano, G. Trappolini, F. Silvestri, DanteLLM: Let's push Italian LLM research forward!, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 4343–4355. URL: https://aclanthology.org/2024.lrec-main.388.

[11] A. Santilli, E. Rodolà, Camoscio: an italian instruction-tuned llama, 2023. arXiv:2307.16456.

[12] D. Groeneveld, I. Beltagy, P. Walsh, A. Bhagia, R. Kinney, O. Tafjord, A. H. Jha, H. Ivison, I. Magnusson, Y. Wang, S. Arora, D. Atkinson, R. Authur, K. R. Chandu, A. Cohan, J. Dumas, Y. Elazar, Y. Gu, J. Hessel, T. Khot, W. Merrill, J. Morrison, N. Muennighoff, A. Naik, C. Nam, M. E. 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. A. Smith, H. Hajishirzi, Olmo: Accelerating the science of language models, 2024. URL: https://arxiv.org/abs/2402.00838. arXiv:2402.00838.

[13] S. Mehta, M. H. Sekhavat, Q. Cao, M. Horton, Y. Jin, C. Sun, I. Mirzadeh, M. Najibi, D. Belenko, P. Zatloukal, M. Rastegari, Openelm: An efficient language model family with open training and inference framework, 2024. URL: https://arxiv.org/abs/2404.14619. arXiv:2404.14619.

[14] T. Nguyen, C. V. Nguyen, V. D. Lai, H. Man, N. T. Ngo, F. Dernoncourt, R. A. Rossi, T. H. Nguyen, Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages, 2023. arXiv:2309.09400.

[15] T. Computer, Redpajama: an open dataset for training large language models, 2023. URL: https:

//github.com/togethercomputer/RedPajama-Data.

[16] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, J. Launay, The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only, arXiv preprint arXiv:2306.01116 (2023). URL: https://arxiv.org/abs/2306.01116. arXiv:2306.01116.

[17] M. Faysse, P. Fernandes, N. M. Guerreiro, A. Loison, D. M. Alves, C. Corro, N. Boizard, J. Alves, R. Rei, P. H. Martins, A. B. Casademunt, F. Yvon, A. F. T. Martins, G. Viaud, C. Hudelot, P. Colombo, Croissantllm: A truly bilingual french-english language model, 2024. URL: https://arxiv.org/abs/2402.00786. arXiv:2402.00786.

[18] G. Penedo, H. Kydlíček, L. B. allal, A. Lozhkov, M. Mitchell, C. Raffel, L. V. Werra, T. Wolf, The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL: https://arxiv.org/abs/2406.17557. arXiv:2406.17557.

[19] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. A. Cosgrove, C. D. Manning, C. Re, D. Acosta-Navas, D. A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. WANG, K. Santhanam, L. Orr, L. Zheng, M. Yuksekgonul, M. Suzgun, N. Kim, N. Guha, N. S. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. A. Chi, S. M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Zhang, Y. Koreeda, Holistic evaluation of language models, Transactions on Machine Learning Research (2023). URL: https://openreview.net/forum?id=iO4LZibEqW, featured Certification, Expert Certification.

[20] A. Lozhkov, R. Li, L. B. Allal, F. Cassano, J. Lamy-Poirier, N. Tazi, A. Tang, D. Pykhtar, J. Liu, Y. Wei, T. Liu, M. Tian, D. Kocetkov, A. Zucker, Y. Belkada, Z. Wang, Q. Liu, D. Abulkhanov, I. Paul, Z. Li, W.-D. Li, M. Risdal, J. Li, J. Zhu, T. Y. Zhuo, E. Zheltonozhskii, N. O. O. Dade, W. Yu, L. Krauß, N. Jain, Y. Su, X. He, M. Dey, E. Abati, Y. Chai, N. Muennighoff, X. Tang, M. Oblokulov, C. Akiki, M. Marone, C. Mou, M. Mishra, A. Gu, B. Hui, T. Dao, A. Zebaze, O. Dehaene, N. Patry, C. Xu, J. McAuley, H. Hu, T. Scholak, S. Paquet, J. Robinson, C. J. Anderson, N. Chapados, M. Patwary, N. Tajbakhsh, Y. Jernite, C. M. Ferrandis, L. Zhang, S. Hughes, T. Wolf, A. Guha, L. von Werra, H. de Vries, Starcoder 2 and the stack v2: The next generation, 2024. arXiv:2402.19173.

[21] Y. Elazar, A. Bhagia, I. H. Magnusson, A. Ravichander, D. Schwenk, A. Suhr, E. P. Walsh, D. Groeneveld, L. Soldaini, S. Singh, H. Hajishirzi, N. A. Smith, J. Dodge, What's in my big data?, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=RvfPnOkPV4.

[22] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, J. Launay, The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only, 2023. URL: https://arxiv.org/abs/2306.01116. arXiv:2306.01116.

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

[24] R. Child, S. Gray, A. Radford, I. Sutskever, Generating long sequences with sparse transformers, arXiv preprint arXiv:1904.10509 (2019).

[25] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, arXiv preprint arXiv:2004.05150 (2020).

[26] G. Gerganov, llama.cpp: Inference of meta's llama model (and others) in pure c/c++, ???? URL: https://github.com/ggerganov/llama.cpp.

[27] T. Dao, FlashAttention-2: Faster attention with better parallelism and work partitioning, in: International Conference on Learning Representations (ICLR), 2024.

[28] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, I. Stoica, Efficient memory management for large language model serving with pagedattention, in: Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles, 2023.

[29] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[30] L. Moroni, S. Conia, F. Martelli, R. Navigli, ITA-Bench: Towards a more comprehensive evaluation for Italian LLMs, in: CLiC-it, 2024.

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

[32] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, K. Toutanova, Boolq: Exploring the surprising difficulty of natural yes/no questions, arXiv preprint arXiv:1905.10044 (2019).

[33] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al., Training verifiers to solve math word problems, arXiv preprint arXiv:2110.14168 (2021).

[34] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, Y. Choi,

Hellaswag: Can a machine really finish your sentence?, arXiv preprint arXiv:1905.07830 (2019).
[35] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, J. Steinhardt, Measuring massive multitask language understanding, Proceedings of the International Conference on Learning Representations (ICLR) (2021).
[36] Y. Bisk, R. Zellers, J. Gao, Y. Choi, et al., Piqa: Reasoning about physical commonsense in natural language, in: Proceedings of the AAAI conference on artificial intelligence, volume 34, 2020, pp. 7432–7439.
[37] J. Welbl, N. F. Liu, M. Gardner, Crowdsourcing multiple choice science questions, arXiv preprint arXiv:1707.06209 (2017).
[38] S. Lin, J. Hilton, O. Evans, Truthfulqa: Measuring how models mimic human falsehoods, arXiv preprint arXiv:2109.07958 (2021).
[39] K. Sakaguchi, R. L. Bras, C. Bhagavatula, Y. Choi, Winogrande: An adversarial winograd schema challenge at scale, Communications of the ACM 64 (2021) 99–106.
[40] F. Mercorio, M. Mezzanzanica, D. Potertì, A. Serino, A. Seveso, Disce aut deficere: Evaluating llms proficiency on the INVALSI Italian benchmark, 2024. URL: https://arxiv.org/abs/2406.17535. arXiv:2406.17535.
[41] X. V. Lin, T. Mihaylov, M. Artetxe, T. Wang, S. Chen, D. Simig, M. Ott, N. Goyal, S. Bhosale, J. Du, R. Pasunuru, S. Shleifer, P. S. Koura, V. Chaudhary, B. O'Horo, J. Wang, L. Zettlemoyer, Z. Kozareva, M. Diab, V. Stoyanov, X. Li, Few-shot learning with multilingual generative language models, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 9019–9052. URL: https://aclanthology.org/2022.emnlp-main.616. doi:10.18653/v1/2022.emnlp-main.616.
[42] S. Mehta, M. H. Sekhavat, Q. Cao, M. Horton, Y. Jin, C. Sun, I. Mirzadeh, M. Najibi, D. Belenko, P. Zatloukal, M. Rastegari, OpenELM: An Efficient Language Model Family with Open Training and Inference Framework, arXiv.org (2024). URL: https://arxiv.org/abs/2404.14619v1.
[43] G. Sarti, M. Nissim, It5: Large-scale text-to-text pretraining for italian language understanding and generation, arXiv preprint arXiv:2203.03759 (2022).
[44] N. Landro, I. Gallo, R. La Grassa, E. Federici, Two new datasets for italian-language abstractive text summarization, Information 13 (2022) 228.
[45] Z. Fu, W. Lam, Q. Yu, A. M.-C. So, S. Hu, Z. Liu, N. Collier, Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder, arXiv preprint arXiv:2304.04052 (2023).
[46] X. Garcia, Y. Bansal, C. Cherry, G. Foster, M. Krikun, M. Johnson, O. Firat, The unreasonable effectiveness of few-shot learning for machine translation, in: International Conference on Machine Learning, PMLR, 2023, pp. 10867–10878.
[47] N. Goyal, C. Gao, V. Chaudhary, P.-J. Chen, G. Wenzek, D. Ju, S. Krishnan, M. Ranzato, F. Guzmán, A. Fan, The flores-101 evaluation benchmark for low-resource and multilingual machine translation, Transactions of the Association for Computational Linguistics 10 (2022) 522–538.
[48] B. Zhang, P. Williams, I. Titov, R. Sennrich, Improving massively multilingual neural machine translation and zero-shot translation, arXiv preprint arXiv:2004.11867 (2020).
[49] R. Navigli, S. Conia, B. Ross, Biases in large language models: Origins, inventory, and discussion, J. Data and Information Quality 15 (2023) 1–21. URL: https://doi.org/10.1145/3597307. doi:10.1145/3597307.
[50] S. Conia, M. Li, D. Lee, U. Minhas, I. Ilyas, Y. Li, Increasing coverage and precision of textual information in multilingual knowledge graphs, 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. 1612–1634. URL: https://aclanthology.org/2023.emnlp-main.100. doi:10.18653/v1/2023.emnlp-main.100.
[51] S. Conia, D. Lee, M. Li, U. F. Minhas, S. Potdar, Y. Li, Towards cross-cultural machine translation with retrieval-augmented generation from multilingual knowledge graphs, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024. URL: https://arxiv.org/abs/2410.14057.

A. Data sources

Table 6 shows the source of each dataset used to train Minerva in its different sizes. The Tokens column shows the total number of tokens we used from each dataset. Where Table 1 shows more tokens used for training, it means they were resampled from the total in order to reach that number. All these datasets are openly licensed.

Dataset	Tokens	Language	Genre	URL
RedPajama-Data-V2	688B	Italian	Web	https://huggingface.co/datasets/togethercomputer/RedPajama-Data-V2
CulturaX	158B	Italian	Web	https://huggingface.co/datasets/uonlp/CulturaX
Wikipedia	1.3B	Italian	Encyclopedic	https://huggingface.co/datasets/wikimedia/wikipedia
Gutenberg	0.15B	Italian	Books	https://huggingface.co/datasets/manu/project_gutenberg
Wikisource	0.12B	Italian	Books	https://huggingface.co/datasets/wikimedia/wikisource
EurLex	1.6B	Italian	Law	https://huggingface.co/datasets/joelito/eurlex_resources
Gazzetta Ufficiale	1.7B	Italian	Law	https://huggingface.co/datasets/mii-llm/gazzetta-ufficiale
FineWeb	1,076B	English	Web	https://huggingface.co/datasets/HuggingFaceFW/fineweb
CulturaX	330B	English	Web	https://huggingface.co/datasets/uonlp/CulturaX
Wikipedia	5.3B	English	Encyclopedic	https://huggingface.co/datasets/wikimedia/wikipedia
ArXiv	33B	English	Academic	https://huggingface.co/datasets/togethercomputer/RedPajama-Data-1T
Gutenberg	7B	English	Books	https://huggingface.co/datasets/manu/project_gutenberg
StackExchange	22B	English	Forum	https://huggingface.co/datasets/togethercomputer/RedPajama-Data-1T
The Stack V2	201B	Code	Code	https://huggingface.co/datasets/bigcode/the-stack-v2-train-smol-ids

Table 6
Detailed breakdown of each dataset.

B. Dataset Insights

We leveraged the WIMBD[5] library to compute word counts per URL domain on CulturaX. We decided not to do this for RedPajama v2 or FineWeb as their original data already provides token count and other insights into the dataset distribution. Figures 2 and 3 show the aggregation of word counts per domain for Italian and English, respectively.

C. Tokenizer

We trained two tokenizers for Minerva. The first one is shared by the three smaller sizes, 350M, 1B and 3B. It is trained on a mix of 4GB of Italian text data and 4GB of English text data, both from CulturaX. Our objective is to have a balanced vocabulary across the two languages, mirroring the training data. We use the SentencePiece library[6] to train a BPE tokenizer and we apply byte fallback. We set a vocabulary size of 32,768 as a multiple of 8, which is recommended by some GPU architectures.

For the 7B tokenizer, we increase the vocabulary size to account for the inclusion of code data, up to 51,200. We also train a BPE tokenizer[7] with 4GB of English text, 4GB of Italian and 1GB of code. The text data is sampled from the training mix of datasets for the 7B, as reported in Table 1.

D. Model

The Minerva LLM family consists of four models, each sharing the same underlying architecture, i.e., that of Mistral-7B. The models are differentiated by their size, ranging from 350 million parameters of Minerva-350M to 7 billion parameters of the largest model, Minerva-7B. The Minerva family also includes Minerva-1B and Minerva-3B, with 1 billion and 3 billion parameters, respectively. More specifically, the Minerva-7B model is based directly on the Mistral-7B architecture, with the sole modifications being the vocabulary size, which we increase to 51,200 tokens, and the context length, which is set to 4,096 tokens without activating the sliding window attention feature. Hence, Minerva-7B is structured as a decoder-only transformer model, comprising 32 layers. Each layer includes 32 attention heads, where each key-value pair is shared among four queries. Additionally, the model features feed-forward layers with a hidden size of 4096 and an intermediate size of 14336, which is 3.5 times the hidden size. Minerva-3B is a scaled down version of Minerva-7B, and it shares similar features with Mistral-7B, including a maximum context length of 16,384 tokens, sliding window attention spanning 2,048 tokens, and a vocabulary size of 32,768 tokens. To achieve approximately 3 billion parameters, we have reduced the hidden size to 2560 and the intermediate size to 8960. Minerva-1B and Minerva-350M differ from their larger counterpart in several key respects. Both models have 16 attention heads, in contrast to the higher count in the larger model. Additionally, the hidden and intermediate sizes of the feed-forward layers is reduced further: Minerva-1B features a hidden size of 2048 and an intermediate size of 7168, while Minerva-350M has a hidden size of 1152 and an intermediate size of 4032. The complete list of parameters is reported in Table 3.

E. News Summarization

Additional results. Table 8 reports the full results of our evaluation on news summarization.

[5] https://github.com/allenai/wimbd
[6] https://github.com/google/sentencepiece
[7] https://huggingface.co/docs/tokenizers/en/api/trainers

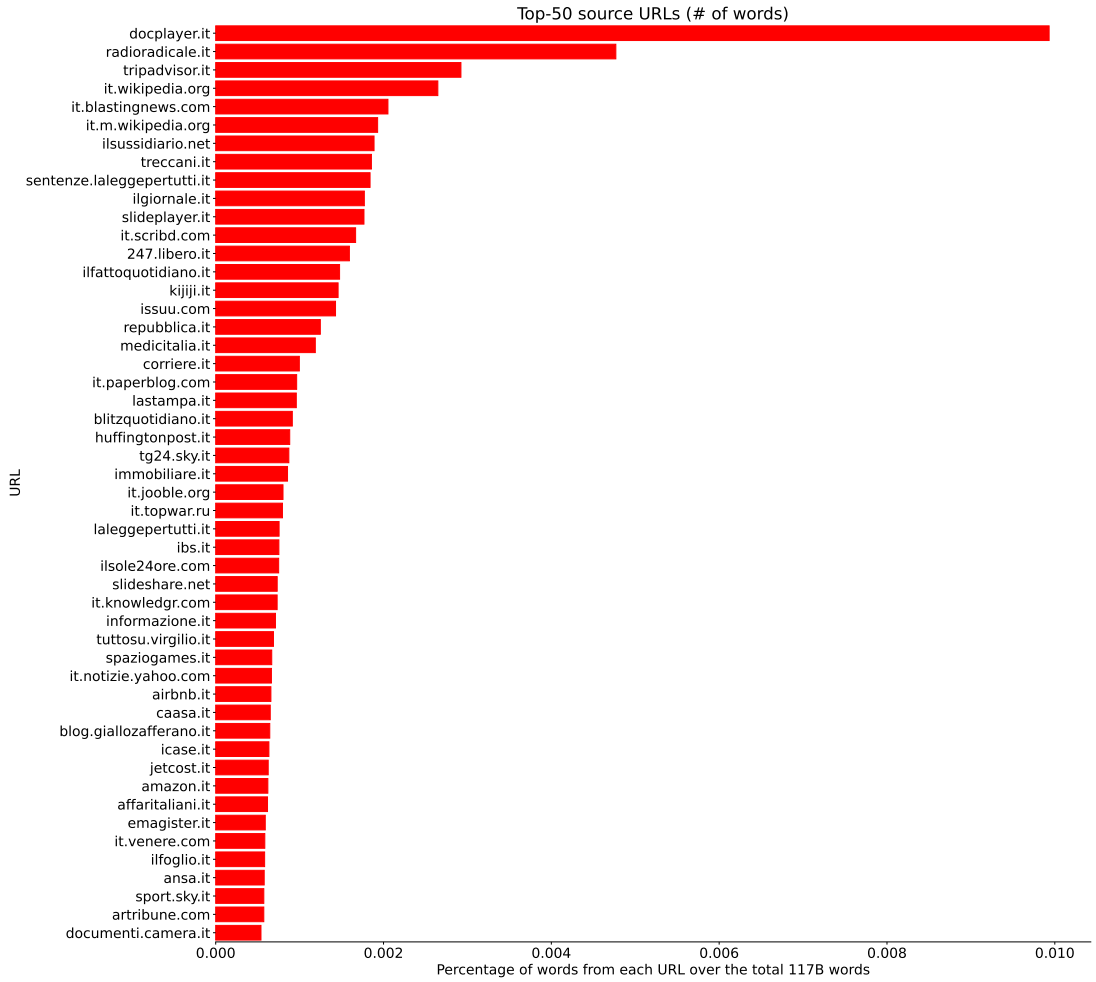

Figure 2: Domain word count distribution for Italian CulturaX.

Model	Optimizer	lr	betas	eps	Weight Decay	Scheduler	Warm-up	Batch Size	Steps
Minerva-350M	AdamW	2×10^{-4}	$(0.9, 0.95)$	10^{-8}	0.0	Cosine	2%	$4M$	$16,690$
Minerva-1B	AdamW	2×10^{-4}	$(0.9, 0.95)$	10^{-8}	0.0	Cosine	2%	$4M$	$47,684$
Minerva-3B	AdamW	2×10^{-4}	$(0.9, 0.95)$	10^{-8}	0.0	Cosine	2%	$4M$	$157,357$
Minerva-7B	AdamW	3×10^{-4}	$(0.9, 0.95)$	10^{-5}	0.1	Cosine	2000	$4M$	$591,558$

Table 7
Training configuration for various Minerva models.

Additional details on the experimental setup. To finetune our Minerva models we relied on the SFTTrainer class.[8] The hyperparameters we used are reported in Table 9. We sought to be in-line with the decisions taken in [43]. We also tried out different combinations, but we noticed that the best evaluation scores are given by the reported parameters. Furthermore, we want to highlight that Minerva-350M and Minerva-1B were finetuned using *AdamW* optimizer [29]. Minerva-3B was trained using *AdamW_Paged_32bit*, a lighter version of AdamW, which allows a larger batch size to be used during training.

[8] https://huggingface.co/docs/trl/en/sft_trainer

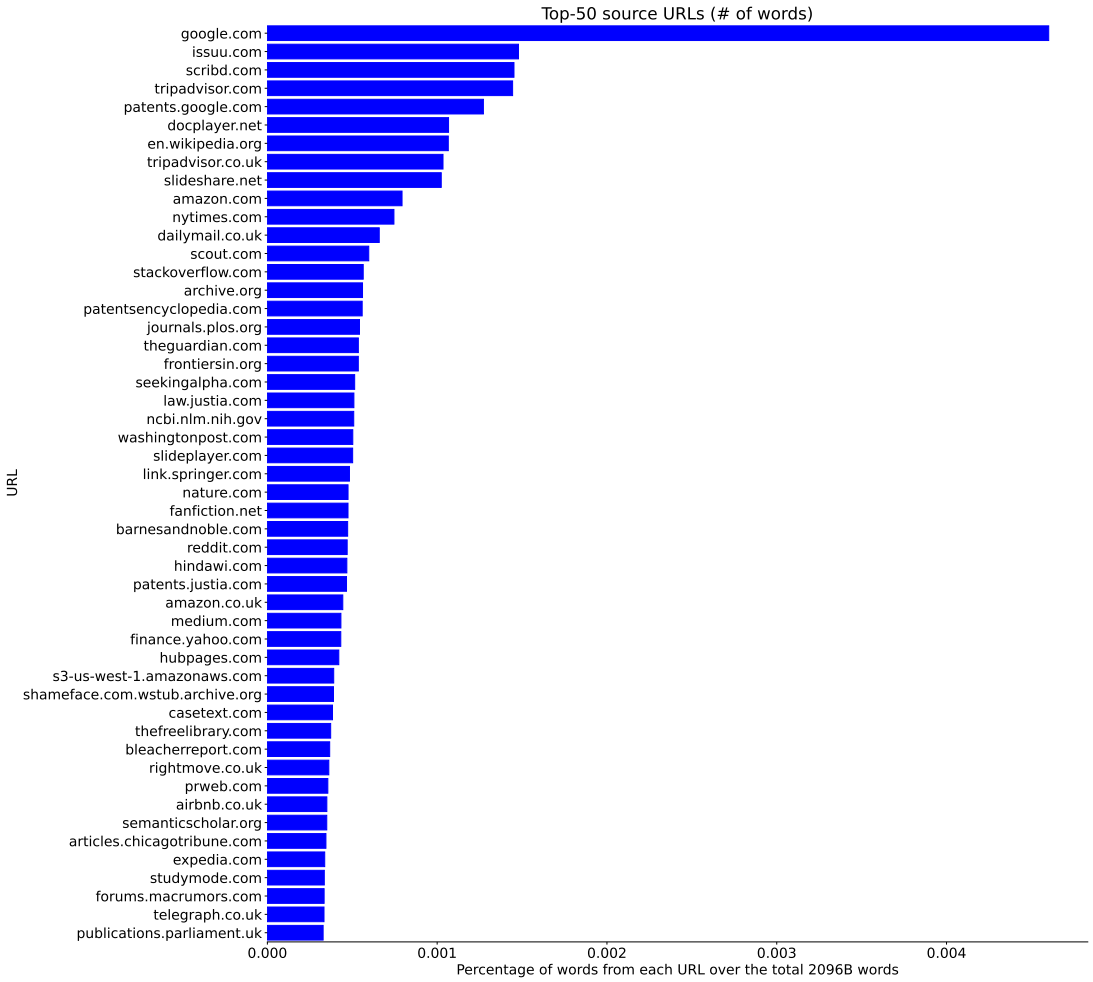

Figure 3: Domain word count distribution for English CulturaX.

F. Few-shot Machine Translation

Here, we provide more details on our experimental setup for the Machine Translation task. In our experiments, we test the capability of a base model (i.e., with no instruction fine-tuning or task-specific fine-tuning) to translate a sentence from English to Italian and vice versa. Previously, LLMs have been shown to perform well in machine translation and they now rival task-specific MT systems on a number of benchmarks [50] and tasks [51]. In our case, we prompt the language models by providing a set of 5 randomly sampled English-to-Italian translations (and vice-versa for the Italian-to-English translation). Finally, we measure the translation performance of the models using COMET, a learned metric to assess the quality between an automatic translation and a gold reference, as COMET has shown better correlation with human judgement than other metrics, such as BLEU.

Model	R1 ↑	R2 ↑	RL ↑
mBART Large	0.32	0.15	0.25
mT5 Small	**0.34**	**0.16**	**0.26**
mT5 Base	0.33	0.16	**0.26**
IT5 Small	0.35	0.17	0.28
IT5 EL32	0.34	0.16	0.26
IT5 Base	0.25	0.10	0.20
IT5 Large	**0.38**	**0.19**	**0.29**
Minerva-350M	0.35	0.17	0.27
Minerva-1B	0.35	0.17	0.27
Minerva-3B	**0.39**	**0.20**	**0.30**

Table 8
Rouge metrics of News Summarization fine-tuning.

Parameter	Value
warmup ratio	0.2
weight decay	5×10^{-3}
batch size	64
optimizer	AdamW | PagedAdamW 32bit (only 3B)
learning rate	0.0005
scheduler	Linear
epochs	7

Table 9
Hyper-parameters used to fine-tune our models.